ORACLE
University

UNIX and Linux Essentials

Activity Guide

D1101651GC10

Table of Contents

For Instructor Use Only.
This document should not be distributed.

Course Practice Environment: Security Credentials

Course Practice Environment: Security Credentials

For operating system (OS) usernames and passwords:

- If you are attending a classroom-based or live virtual class (LVC), ask your instructor or LVC producer for OS credentials (`oracle` and `root` user).

- If you are using a self-study format, refer to the communication that you received from Oracle University for this course.

For product-specific credentials used in this course, see the following table:

Product-Specific Credentials		
Virtual Machines/Application	**Username**	**Password**
host01/OS	oracle	oracle
host01/OS	root	oracle
host02/OS	oracle	oracle
host02/OS	root	oracle

Practices for Lesson 1:
Course Introduction

Practices Overview

In this practice, you will be introduced to your practice environment.

Practice 1-1: An Introduction to Your Practice Environment

Overview

The course practice environment consists of two Oracle Linux systems, `host01` and `host02`, that can communicate with each other. You will perform all the practices in the `host01` system, unless specified otherwise.

Assumptions

You are logged on as the `oracle` user on the `host01` system.

Tasks

1. Ensure you can see the Oracle Linux desktop.
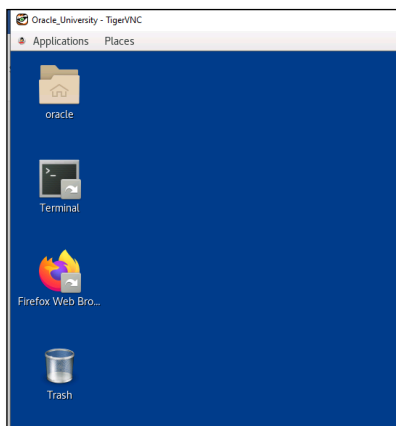
    Note: Some elements of your desktop, such as the background, colors, or additional displayed shortcuts, may vary.

2. Know your present working directory.

    a. Double-click the **Terminal** icon to open a terminal window.

    b. Type **pwd** and press the **Enter** key.

    ```
    [oracle@host01]$ pwd
    /home/oracle
    [oracle@host01]$
    ```

3. To become the `root` user, in the terminal window, type the **`su -`** command to assume primary administrator privileges.

   a. Type `su -` and press the **Enter** key.

   b. Enter the password for the `root` user. (See *Course Practice Environment: Security Credentials* for the `root` user password.)

   ```
   [oracle@host01]$ su -
   Password: <Password>
   Last login: Wed Nov 25 09:54:14 UTC 2020 on pts/1
   [root@host01]#
   ```

   Note: When entering the password, it will not be displayed, and the prompt will change from a dollar sign (`$`) to a hash (`#`), indicating you are logged in as a privileged user.

4. Close the terminal window.

   Type **exit** and press the **Enter** key to close the terminal window.

   ```
   [root@host01]# exit
   ```

5. Explore Lab files.

   a. Double-click the **oracle** icon on the desktop.
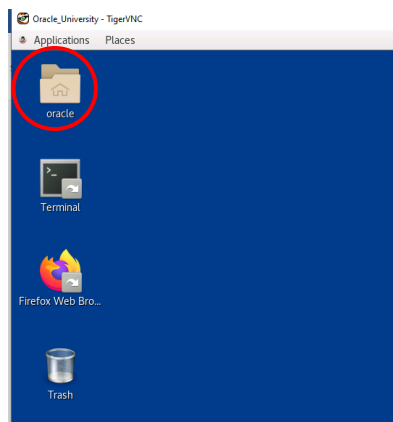
   b. Double-click the **lab** folder. The folder path is `/home/oracle/lab`. This folder contains the required resource files to implement practice instructions for this course.

   Note: Similarly, you can log on and explore the `host02` system.

**Practices for Lesson 2:
Introduction to the UNIX and
Linux Environments**

Practices for Lesson 2: Overview

Practices Overview

In these practices, you will:

- Log on to the system and change your user login password
- Display system information using the command line
- Use the man pages

Practice 2-1: Logging On to the System and Changing Your User Login Password

Overview

In this practice, you will learn to change the user password, and use a terminal window accessed from the desktop login window.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)

Tasks

1. Change your user password.

   Oracle Linux follows a strict default password authentication mechanism for regular users. As you are logged on as a regular user, the `oracle` user, you must set a strong password that conforms to strict password requirements. However, if you change to the `root` user and set a password, the system will accept any password.

   Only for the following steps, you will switch to the `root` user to set the password for the `oracle` user, and later return it to its original setting.

   a. Open a terminal window.

      To open a terminal window, right-click the desktop and select the **Open Terminal** option.

   b. Change to the `root` user by using the `su -` command and input the `root` password as `oracle1`, when prompted. (See *Course Practice Environment: Security Credentials* for the `root` user password).

      ```
      [oracle@host01]$ su -
      Password: <Password>
      Last login: Thu Nov 26 06:18:08 UTC 2020 on pts/0
      [root@host01]#
      ```

   c. Use the `passwd oracle` command to change your `oracle` user password to `mypass1`. When the password is set, use the `exit` command to return to the `oracle` user.

      Note: A `BAD PASSWORD` warning is displayed that the password fails policy requirements, but as the `root` user, the password is accepted after you input it a second time.

      ```
      [root@host01]# passwd oracle
      Changing password for user oracle.
      New password: mypass1
      BAD PASSWORD: The password is shorter than 8 characters
      Retype new password: mypass1
      ```

```
passwd: all authentication tokens updated successfully.
[root@host01]#
```

d. Use the `exit` command to switch back to the `oracle` user. Close the terminal window by entering `exit` again.

```
[root@host01]# exit
logout
[oracle@host01]$ exit
```

2. On the desktop environment, click the **down-arrow** icon in the top-right corner.
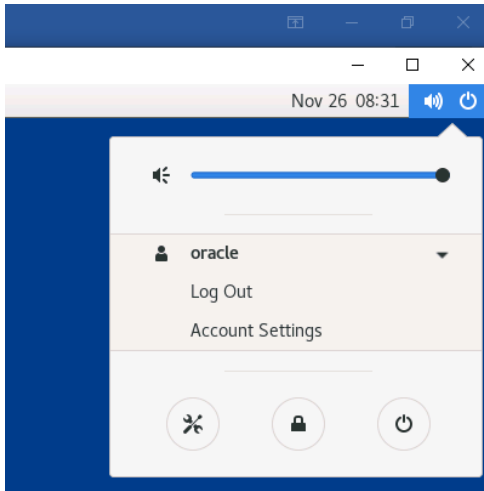
a. Click **oracle** and then click **Log Out** to log out of the desktop environment. A logout confirmation window appears.
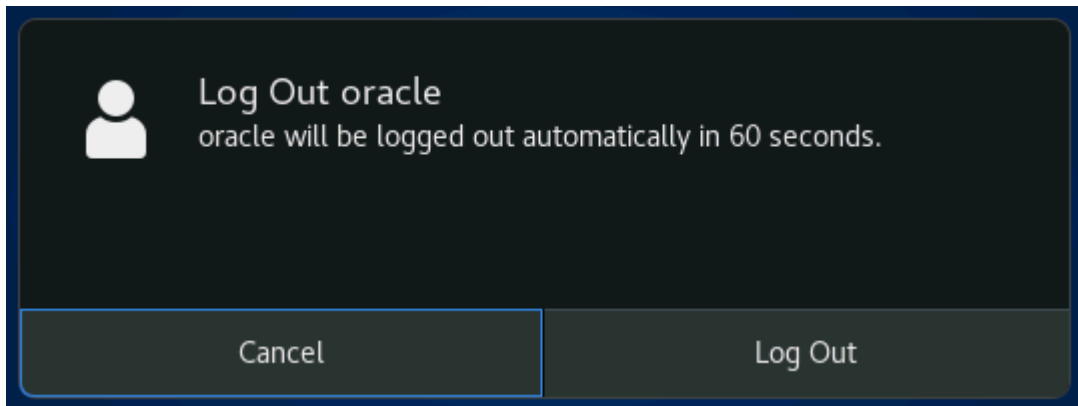
b. Click **Log Out**.

c. Access the `host01` system again.

.
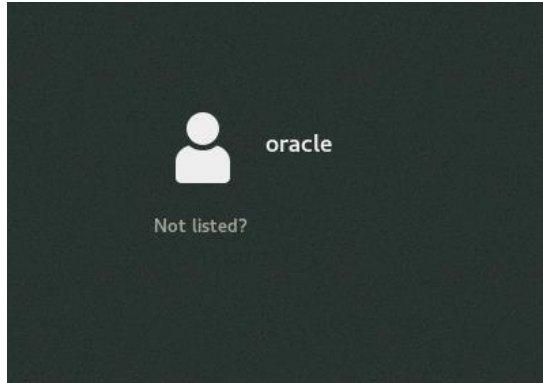
d. Click **Not listed?** to try an incorrect username and password. The **Username** screen appears.
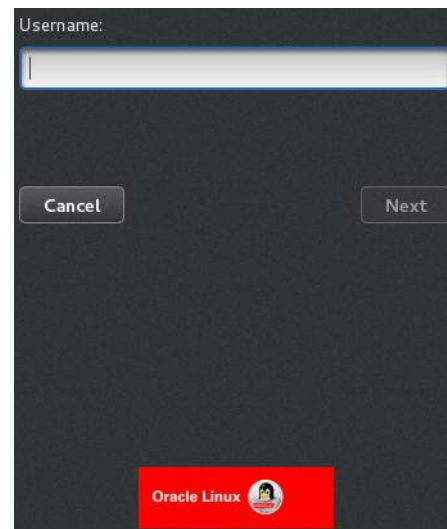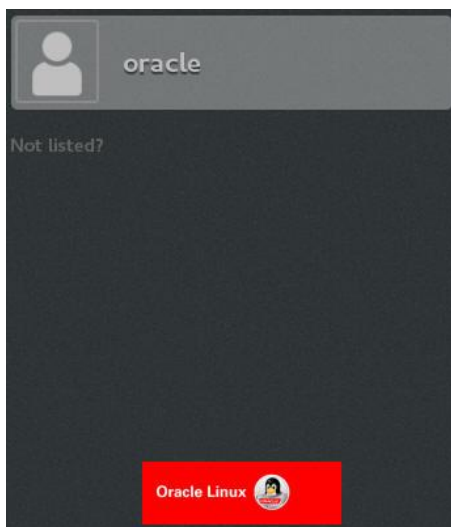
e. Enter the following incorrect username and password on the Username screen:

- Username: **oracle2**

- Password: **wrong**

The following dialog box appears indicating authentication failure. Click **Cancel** or press **Enter.** The login screen reappears.

Password:

|

Sorry, that didn't work. Please try again.

Cancel   ⚙ Sign In
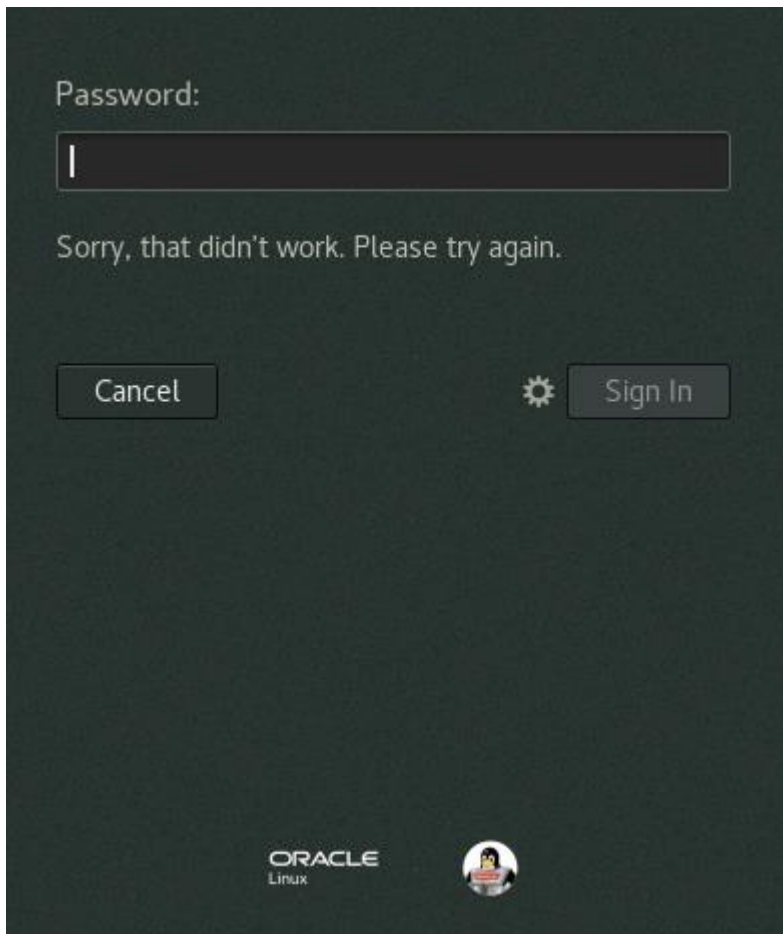
ORACLE
Linux

f. Log in with the correct user credentials.

- Username: **oracle**

- Password: **mypass1**

Reset the oracle user password to its default setting of oracle. Open a terminal window and switch to the root user with the su – command. Use the passwd oracle command to set the password to oracle. Exit the terminal window and log out of the desktop. Log back in using the username oracle and password oracle. After verifying that the reset password is functioning, log out of the desktop again.

Practice 2-2: Displaying System Information Using the Command Line

Overview

In this practice, you will display information about the OS and the system.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)

Tasks

1. Open a terminal window.

   To open a terminal window, right-click the open/free area of the Oracle Linux desktop and select the **Open Terminal** option.

2. Display information about the OS and the system.

   ```
   [oracle@host01]$ uname -a
   Linux host01 5.4.17-2011.5.3.el8uek.x86_64 #2 SMP Wed Jul 29
   22:09:11 PDT 2020 x86_64 x86_64 x86_64 GNU/Linux
   [oracle@host01]$
   ```

3. Display information about the OS name.

   ```
   [oracle@host01]$ uname -s
   Linux
   [oracle@host01]$
   ```

4. Display information about the OS release level.

   ```
   [oracle@host01]$ uname -r
   5.4.17-2011.5.3.el8uek.x86_64
   [oracle@host01]$
   ```

5. Display the current date and time.

   ```
   [oracle@host01]$ date
   Thu Nov 26 08:57:17 UTC 2020
   [oracle@host01]$
   ```

6. Display the current month's calendar.

   ```
   [oracle@host01]$ cal
        November 2020
   Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6  7
    8  9 10 11 12 13 14
   15 16 17 18 19 20 21
   22 23 24 25 26 27 28
   29 30
   [oracle@host01]$
   ```

7. Display the calendar with a specific month and year (for example, June 2017).

```
[oracle@host01]$ cal 06 2017
      June 2017
 S  M Tu  W Th  F  S
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30


[oracle@host01]$
```

8. Clear the terminal window using the clear command.

9. Display the current date and host system name.

```
[oracle@host01]$ date; hostname
Thu Nov 26 08:59:01 UTC 2020
host01
[oracle@host01]$
```

10. Display the calendar for March 2017, the current date, and the OS release information.

```
[oracle@host01]$ cal 03 2017; date; cat /etc/system-release
     March 2017
 S  M Tu  W Th  F  S
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31


Thu Nov 26 09:00:14 UTC 2020
Oracle Linux Server release 8.2
[oracle@host01]$
```

Practice 2-3: Using the Man Pages

Overview

In this practice, you will learn to use the `man` command to extract additional information about system commands.

Assumptions

- You are logged on as the `oracle` user on the `host01` system.
- You have opened a terminal window.

Tasks

1. Display the man[ual] pages and man page sections.

   a. To display the manual pages for the `man` command, run the `man man` command.

   ```
   [oracle@host01]$ man man
   MAN(1)                    Manual pager utils                    MAN(1)

   NAME
          man – an interface to the on-line reference manuals

   .. Output truncated ..
   ```

   Viewing man pages using the `less` (pager) keyboard commands:

Keyboard Commands	Functions
`h`	Provides a description (help) of all scrolling capabilities
Space bar	Displays the next screen of a man page
Return/Enter	Displays the next line of a man page
`b`	Moves back one full screen of the man pages
`g`	Returns to the top of the man pages
`G`	Goes to the bottom of the man pages
`/pattern`	Searches forward for a *pattern* (regular expression)
`?pattern`	Searches backward for a *pattern* (regular expression)
`n`	Finds the next occurrence of the *pattern*
`N`	Changes the direction of the search
`q`	Quits the `man` command and returns to the shell prompt

b. To display man pages section 1 (User Commands), which is the default for the `man` command, run the following commands:

```
[oracle@host01]$ man 1 man
:
:
.. Output truncated ..

[oracle@host01]$ man man
```

c. To display man pages section 2 (System Calls) for the `exit` command, run the command as follows:

```
[oracle@host01]$ man 2 exit
```

d. Use the keyboard commands to scroll through the man pages. You can also search for a pattern by entering `/<pattern>`. For example, choose a pattern and search for it in the man pages for the `uname` command.

e. Using the `man -k man` command, search the man pages for information on the `man` command by using the keyword "man".

f. Search the man pages for information on the `passwd` command.

g. Display section 5 (file formats) in Oracle Linux of the man pages for the `passwd` file format and review its contents.

Note: Use the letter `q` key command to quit the `man` command.

Practices for Lesson 3: Working with Files and Directories

Practices for Lesson 3: Overview

Practices Overview

In these practices, you will:

- Access files and directories
- Use file and directory commands
- Locate files and text

Practice 3-1: Accessing Files and Directories

Overview

In this practice, you will use file and directory access commands. You will use the files and directories available in the `/home/oracle/lab` directory.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

1. Open a terminal window by right-clicking the desktop. Select the **Open Terminal** option.

2. Display user information by using the `id` command.

```
[oracle@host01]$ id
uid=1000(oracle) gid=1000(oracle) groups=1000(oracle)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[oracle@host01]$
```

3. Display your current working directory by using the `pwd` command.

```
[oracle@host01]$ pwd
/home/oracle
```

4. Change to your home directory from any location by using the `cd` command.

```
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
```

5. Display the contents of your current working directory by using the `ls` command.

```
[oracle@host01]$ ls
Desktop     Downloads   lab     Pictures   Templates
Documents   eKit        Music   Public     Videos
```

6. Display all files, including any hidden files, using the `ls -a` command.

```
[oracle@host01]$ $ ls -a
                .cache     .esd_auth      Music      Videos
..              .config    .ICEauthority  Pictures   .viminfo
.bash_history   Desktop    lab            .pki       .Xauthority
.bash_logout    Documents  .lesshst       Public
.bash_profile   Downloads  .local         .ssh
.bashrc         eKit       .mozilla       Templates
[oracle@host01]$
```

7. Display a long list of the contents of the current working directory by using the `ls -l` command.

```
[oracle@host01]$ ls -l
total 40
drwxr-xr-x. 2 oracle oracle 4096 Sep  3 14:09 Desktop
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Documents
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Downloads
drwxrwxrwx. 2 root   root   4096 Dec 18 15:03 eKit
drwxr-xr-x. 7 oracle oracle 4096 Dec 18 14:51 lab
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Music
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Pictures
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Public
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Templates
drwxr-xr-x. 2 oracle oracle 4096 Sep  1 20:37 Videos
[oracle@host01]$
```

8. Display the file types in your current working directory by using the `ls -F` command.

```
[oracle@host01]$ ls -F
Desktop/    Downloads/  lab/    Pictures/  Templates/
Documents/  eKit/       Music/  Public/    Videos/
[oracle@host01]$
```

9. Change to the `lab/dir1` directory by using the `cd` command.

```
[oracle@host01]$ cd lab/dir1
[oracle@host01]$ pwd
/home/oracle/lab/dir1
```

10. Display a long list of the contents of the current working directory by using the `ls -l` command.

```
[oracle@host01]$ pwd
/home/oracle/lab/dir1
[oracle@host01]$ ls -l
total 4
drwxr-xr-x. 3 oracle oracle 4096 Mar 29  2018 coffees
```

11. Change to the `coffees` directory by using the `cd` command.

```
[oracle@host01]$ cd coffees
[oracle@host01]$ pwd
/home/oracle/lab/dir1/coffees
```

12. Change to the `planets` directory, which is available under the `$HOME/lab/dir3` directory, by using the relative path name.

```
[oracle@host01]$ cd ../../dir3/planets
[oracle@host01]$ pwd
/home/oracle/lab/dir3/planets
[oracle@host01]$
```

Now, return to the home directory.

```
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$
```

13. Change to the `dir1` directory by using the absolute path name, and then return to your home directory.

```
[oracle@host01]$ cd /home/oracle/lab/dir1
[oracle@host01]$ pwd
/home/oracle/lab/dir1
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$
```

Note: The `cd ~/lab/dir1` command will also work in addition to the command shown.

14. Change to the `/etc` directory by using the relative path name. Then change to the `lab` directory in your home directory, and finally change to the `dir1` directory.

```
[oracle@host01]$ cd ../../etc
[oracle@host01]$ pwd
/etc
[oracle@host01]$ cd ~/lab
[oracle@host01]$ pwd
/home/oracle/lab
[oracle@host01]$ cd dir1
[oracle@host01]$ pwd
/home/oracle/lab/dir1
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$
```

15. Display the contents of the `fruit` file by using the `cat` command with line numbers.

```
[oracle@host01]$ cd lab
[oracle@host01]$ cat -n fruit
 1  lemon
 2  orange
 3  apple
 4  banana
 5  pear
 6  mango
 7  tomato
 8  pomegranate
 9
[oracle@host01]$
```

16. Display the contents of the `fruit` and `fruit2` files by using a single command.

```
[oracle@host01]$ cat fruit fruit2
lemon
orange
apple
banana
pear
mango
tomato
pomegranate

lemon
orange
apple
banana
tomato
guava
mango
pomegranate
[oracle@host01]$
```

17. Display the first five lines of the `/usr/share/dict/words` file on the screen.

```
[oracle@host01]$ head -5 /usr/share/dict/words
1080
10-point
10th
11-point
12-point
[oracle@host01]$
```

18. Display the last eight lines of the `/usr/share/dict/words` file on the screen.

```
[oracle@host01]$ tail -8 /usr/share/dict/words
Zyzomys
Zyzzogeton
zyzzyva
zyssyvas
ZZ
Zz
zZt
ZZZ
[oracle@host01]$
```

Note: The `head` command displays the first 10 lines of a file, The `tail` command displays the last 10 lines of a file.

19. Determine the total number of lines contained in the `/usr/share/dict/words` file by using the `wc -l` command.

```
[oracle@host01]$ wc -l /usr/share/dict/words
479828 /usr/share/dict/words
[oracle@host01]$
```

Practice 3-2: Using File and Directory Commands

Overview

In this practice, you will perform some actions on files and directories using file and directory commands.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

To use file and directory commands, complete the following steps:

1. If you are not in the `lab` subdirectory in your home directory, change to the `lab` subdirectory.

   ```
   [oracle@host01]$ pwd
   /home/oracle
   [oracle@host01]$ cd ~/lab
   [oracle@host01]$ pwd
   /home/oracle/lab
   [oracle@host01]$
   ```

2. From the `/home/oracle/lab` directory, make a new directory, `dir4`, copy the `dir1/coffees/beans/beans` file into the `dir4` directory, and name it `roses`.

   ```
   [oracle@host01]$ mkdir dir4
   [oracle@host01]$ cp dir1/coffees/beans/beans dir4/roses
   [oracle@host01]$ ls dir4
   roses
   [oracle@host01]$
   ```

3. Create a directory called `vegetables` in `dir3`.

   ```
   [oracle@host01]$ mkdir dir3/vegetables
   ```

4. Move the `dir1/coffees/beans/beans` file into the `dir2` directory.

   ```
   [oracle@host01]$ mv dir1/coffees/beans/beans dir2/
   [oracle@host01]$ ls dir2
   beans   notes
   ```

The command options for the `cp`, `mv`, and `rm` commands are described in the following table:

Option	Description
-f	Force. Do not prompt before overwrite or removal of existing files or directories.
-i	Interactive; prompts before accidental overwrite or removal of existing files or directories
-r or -R	Recursive; when working with directories, includes the contents of the directory and all subdirectories
-v	Verbose; explains what is being done

5. From your `lab` directory, create a directory called `practice1`.

```
[oracle@host01]$ pwd
/home/oracle/lab
[oracle@host01]$ mkdir practice1
```

6. Using a single command, copy the `file.1` and `file.2` files into the `practice1` directory.

```
[oracle@host01]$ cp file.1 file.2 practice1
[oracle@host01]$ ls practice1
file.1 file.2
[oracle@host01]$
```

7. Copy the `dir3/planets/mars` file to the `practice1` directory, and name the file `addresses`.

```
[oracle@host01]$ cp dir3/planets/mars practice1/addresses
[oracle@host01]$ ls practice1
addresses file.1 file.2
[oracle@host01]$
```

8. Create a directory called `play` in your `practice1` directory, and move the `practice1/addresses` file to the `play` directory.

```
[oracle@host01]$ mkdir practice1/play
[oracle@host01]$ ls practice1
addresses file.1 file.2 play
[oracle@host01]$ mv practice1/addresses practice1/play
[oracle@host01]$ ls practice1 practice1/play
practice1:
file.1 file.2 play
```

```
practice1/play:
addresses
[oracle@host01]$
```

9. Using a single command, copy the `play` directory in the `practice1` directory to a new directory in the `practice1` directory called `appointments`.

```
[oracle@host01]$ cp -r practice1/play practice1/appointments
[oracle@host01]$
```

10. Recursively list the contents of the `practice1` directory.

```
[oracle@host01]$ ls -R practice1
practice1:
appointments  file.1        file.2        play

practice1/appointments:
addresses

practice1/play:
addresses
[oracle@host01]$
```

11. In your home directory, create a directory called `house` with a subdirectory called `furniture` using a single command.

```
[oracle@host01]$ cd; mkdir -p house/furniture
```

12. Create an empty file called `chairs` in the new `furniture` directory.

```
[oracle@host01]$ touch house/furniture/chairs
```

13. Using a single command, create three directories called `records`, `memos`, and `misc` in your `home` directory.

```
[oracle@host01]$ mkdir records memos misc
```

14. Create a new file called `carrot`, and rename it `celery`.

```
[oracle@host01]$ touch carrot
[oracle@host01]$ mv carrot celery
```

15. Using a single command, remove the directories called `memos` and `misc` from your home directory.

```
[oracle@host01]$ rmdir memos misc
```

Note: A recursive remove can be performed using the `rm -r memos misc` command.

16. Try to remove the directory called `house/furniture` with the `rm` (no options) command. Observe what happens.

```
[oracle@host01]$ rm house/furniture
rm: cannot remove 'house/furniture': Is a directory
[oracle@host01]$
```

Practices for Lesson 3: Working with Files and Directories

17. Use the `rm -r` command to remove a directory that is not empty. Remove the `house/furniture` directory. List the contents of the `house` directory to verify that the `furniture` directory has been removed.

```
[oracle@host01]$ rm -r house/furniture
[oracle@host01]$ ls house
[oracle@host01]$
```

18. Create a new directory named `newname`, and rename it `veggies`.

```
[oracle@host01]$ mkdir newname
[oracle@host01]$ mv newname veggies
[oracle@host01]$ ls newname veggies
ls: cannot access newname: No such file or directory
veggies:
[oracle@host01]$
```

19. Create a symbolic link called `myprofile` that is a symbolic link to the `/etc/profile` file.

```
[oracle@host01]$ ln -s /etc/profile myprofile
[oracle@host01]$ ls -l myprofile
lrwxrwxrwx ... myprofile -> /etc/profile
[oracle@host01]$
```

20. Verify that the symbolic link works.

```
[oracle@host01]$ head -9 myprofile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what
you
# are doing.  It's much better to create a custom.sh shell
script in
# /etc/profile.d/ to make custom changes to your environment, as
this
# will prevent the need for merging in future updates.
[oracle@host01]$
```

21. Remove the symbolic link previously created.

```
[oracle@host01]$ rm myprofile
[oracle@host01]$ ls myprofile
ls: cannot access myprofile: No such file or directory
[oracle@host01]$
```

Practice 3-3: Locating Files and Text

Overview

In this practice, you will use `grep` and related commands to locate files and text in files.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Before You Begin

As a reminder for these tasks, note the following for your reference:

- The `grep` command searches the contents of one or more files for a character pattern using full regular expression metacharacters.
- The `egrep` (`grep -E`) command searches the contents of one or more files for one or more patterns using **extended** regular expression metacharacters.
- The `fgrep` (`grep -F`) command searches a file for a literal (**fixed**) string or a group of characters.

Tasks

1. Search for the text string `root` in the `/etc/group` file and display it on the screen.

```
[oracle@host01]$ grep root /etc/group
root:x:0:
[oracle@host01]$
```

2. In the `lab` directory under your home directory, display all lines in the `dante`, `file1`, and `dante_1` files that contain the lowercase pattern "who".

```
[oracle@host01]$ cd lab
[oracle@host01]$ grep who dante file1 dante_1
dante:Mention "Alighieri" and fee will know whom you are talking
about, say
dante:"Dante," instead, and the whole world knows whom you mean.
For
dante:Who is this Dante, whom T.S. Eliot calls "the most
universal of poets
file1:With others within their own depts, other depts, and the
whole organization.
dante_1:Santa Croce, and later at Bologna with Brunetto Latini,
who taught him,
... Output truncated ...
[oracle@host01]$
```

Note: Boldface added for illustration.

3. Use the `grep` command to look for all lines in the `file4` file that do not contain the uppercase letter `M`.

```
[oracle@host01]$ cat -n file4
    1 The budget for quarter 2 is:
    2
    3 Marketing.............8M
    4 Sales................12M
    5 Finance..............4.5M
    6 Manufacturing........1M
[oracle@host01]$ grep -v M file4
The budget for quarter 2 is:
[oracle@host01]$
```

4. Use the `egrep` command to display all lines in the `file4` file that contain either the "`Sales`" or the "`Finance`" pattern.

```
[oracle@host01]$ egrep '(Sales|Finance)' file4
Sales................12M
Finance..............4.5M
[oracle@host01]$
```

5. Display all the lines that have the pattern "`kernel`" in the `/usr/lib/sysctl.d/50-default.conf` file with line numbers.

```
[oracle@host01]$ grep -n kernel /usr/lib/sysctl.d/50-default.conf
14:# System Request functionality of the kernel (SYNC)
16:# Use kernel.sysrq = 1 to allow all keys.
17:# See https://www.kernel.org/doc/html/latest/admin-guide/sysrq.html
for a list
19:kernel.sysrq = 16
22:kernel.core_uses_pid = 1
25:kernel.kptr_restrict = 1
[oracle@host01]$
```

Note: Boldface added for illustration.

6. Use the `grep` command to display the number of lines that contain the pattern "`net`" in the `/usr/lib/sysctl.d/50-default.conf` file.

```
[oracle@host01]$ grep -c net /usr/lib/sysctl.d/50-default.conf
4
[oracle@host01]$
```

7. Starting with your home directory, find all files of type `f`.

```
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ find ~ -type f
/home/oracle/.bash_logout
/home/oracle/.bash_profile
/home/oracle/.bashrc
/home/oracle/.cache/gdm/session.log.old
/home/oracle/.cache/gdm/session.log
/home/oracle/.cache/imsetings/log.bak
/home/oracle/.cache/imsetings/log


... Output truncated ...
```

8. Starting in your home directory, find all files of type `d` for directory.

```
[oracle@host01]$ find ~ -type d
/home/oracle
/home/oracle/.mozilla
/home/oracle/.mozilla/extensions
/home/oracle/.mozilla/plugins
/home/oracle/.cache/
/home/oracle/.cache/gdm


... Output truncated ...
```

9. Starting in your home directory, find all the files that contain the pattern "*dante*".

```
[oracle@host01]$ find . -name *dante*
./lab/dante
./lab/dante_1
[oracle@host01]$
```

10. Starting in your home directory, find all the files that were modified in the last one day.

```
[oracle@host01]$ find . -mtime -1
.
./.cache/tracker/meta.db-wal
./.cache/tracker/meta.db.shm
./.config/dconf
./.config/dconf/user
./.local/share/tracker/data/tracker-store.journal


... Output truncated ...
```

11. From your home directory, use the `find` command to search for ordinary files of size 0 (zero), beginning in your `lab` directory. Include an option prompting you with `yes` or `no` before long-listing the files.

```
[oracle@host01]$ find lab -type f -size 0 -ok ls -l {} \;
< ls ... lab/dir2/notes > ? yes
-rw-r--r--. 1 oracle oracle 0 Mar  5 17:36 lab/dir2/notes
< ls ... lab/file.3 > ? yes
-rw-r--r--. 1 oracle oracle 0 Mar  5 17:36 lab/file.3
< ls ... lab/file.2 > ? no
< ls ... lab/file.1 > ? ^C
[oracle@host01]$
```

Note: To break out of the output, press Ctrl + C (^C).

Practices for Lesson 4: Using the vim Editor

Practices for Lesson 4: Overview

Practices Overview

In these practices, you will use the `vim` editor.

Practice 4-1: Using the `vim` Editor

Overview

In this practice, you will use the `vimtutor` executable file. The `vim` editor is the improved version of the `vi` editor. `vim` is the default editor.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

1. Before starting `vimtutor`, make sure that you are in your home directory.

```
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ which vimtutor
/usr/bin/vimtutor
[oracle@host01]$ vimtutor
```

Note: When you start `vimtutor`, it makes a copy of the `vim` tutor file so that the original file is protected against modifications. This tutor is set up to teach by use. To quit `vimtutor`, press the **Esc** key to return to command mode, and then enter `:q!` to quit the tutor and return to the system prompt.

2. Show that, in Oracle Linux, `vi` is an alias of `vim`.

```
[oracle@host01]$ which vi
alias vi='vim'
        /usr/bin/vim
[oracle@host01]$
```

3. In your `/home/oracle` directory, create a file called `example`. The terminal window screen is replaced by the `vi` interface. At the top of the terminal window is the blinking cursor. At the bottom of the terminal window, you see "example" [New File].

```
[oracle@host01]$ vi example
~
~
... Output truncated ...
"example" [New File]                                    0,0-1    All
```

Note: To move to the next line to insert the sentence, press **i** and **Enter**.

4. Press the **i** key to change into insert mode and insert the following text:

```
Hello World
What is your
Waht id today's date?
```

```
Hello World
What is your
Waht id today's date?
~
~
~
-- INSERT --
```

5. To append text to the line `What is your`, press **Esc** to enter command mode. Use the **h**, **j**, **k**, **l**, or **arrow** keys to navigate to the last character of the line. Press the **a** key to append and insert a space with the next string "`name?`".

```
Hello World
What is your name?
Waht id today's date?
~
~
-- INSERT --
```

6. To replace the `d` character with `s` in the line, `Waht id today's date?`, press **Esc** to return to command mode. Then move the cursor to the third line by pressing the **j** or **down arrow** key. This will move the cursor down. To move the cursor to the left, press **h** or the **left arrow** key. Bring the cursor to the `d` character in the string "`id`". Press the **r** key and then insert character `s`. This will replace the character `d` with the character `s`.

```
Hello World
What is your name?
Waht is today's date?
~
~
~
```

Note: Ensure that you are in command mode before you press the **r** command key.

7. To change the word Waht to What, press **Esc** and move the cursor to the third line. Place your cursor on the character "a" of the word Waht and execute the **cw** command. Enter the text hat. This will change the whole word Waht to what. Press **Esc** when you finish modifying the word.

```
Hello World
What is your name?
What is today's date?
~
~
~
```

8. To copy and paste the line Hello World, press **Esc** to return to command mode. Move the cursor to the beginning of the Hello World line. Execute the **yy** command to copy the string. Then move the cursor to the end of the same line and execute the **p** command to paste the string. The whole line is copied and pasted.

```
Hello World
Hello World
What is your name?
What is today's date?
~
~
~
```

Note: Ensure you are in command mode before executing the **yy** and **p** commands.

9. To delete the additional "Hello World" line, press **Esc** to enter command mode. Move the cursor to the beginning of the second line "Hello World" and execute the **dd** command. The entire line is deleted.

```
Hello World
What is your name?
What is today's date?
~
~
~
```

Note: Ensure you are in command mode before you execute the **dd** command.

10. To search for the string "`What`", press **Esc** to enter command mode and press the forward slash **/** key. Enter the text "`What`" and press **Enter**. The cursor automatically moves to the first string in the file that it encounters. Notice that "`/What`" appears at the bottom of the terminal window screen.

```
Hello World
What is your name?
What is today's date?
~
~
~
/What
```

Note: The matching strings found are also highlighted in yellow.

11. To search for the next occurrence of the same string, press **n**. The cursor will move to the second string in the file.

```
Hello World
What is your name?
What is today's date?
~
~
~
search hit BOTTOM, continuing at TOP
```

12. To customize the session by displaying the line numbers, press **Esc** to enter command mode. Then enter the `:set nu` command and press **Enter**. Notice that `:set nu` appears at the bottom of the terminal window screen.

```
Hello World
What is your name?
What is today's date?
~
~
:set nu
```

The output changes to:

```
      1 Hello World
      2 What is your name?
      3 What is today's date?
~
~
~
:set nu
```

13. To remove the line numbers, press **Esc** to enter command mode. Next, type the `:set nonu` command and press **Enter**. The line numbers disappear.

```
Hello World
What is your name?
What is today's date?
~
~
~
:set nonu
```

14. To quit and save the file with the changes, press **Esc** to enter command mode. Then type `:wq` and press **Enter**. Notice that `:wq` appears at the bottom of the terminal window screen. The file is saved and the command prompt returns.

```
Hello World
What is your name?
What is today's date?
~
~
:wq
[oracle@host01]$
```

15. For more information about the various commands in `vim`, refer to the `vimtutor` executable file you used at the beginning of this practice.

Practices for Lesson 4: Using the vim Editor

Practices for Lesson 5: Using Features Within the Bash Shell

Practices for Lesson 5: Overview

Practices Overview

In these practices, you will:

- Use shell metacharacters
- Use command redirection
- Use variables in the Bash shell
- Display the command history
- Customize the user's work environment

Practice 5-1: Using Shell Metacharacters

Overview

In this practice, you will use shell metacharacters to simplify commands, structure, and output. `bash` is the default shell in Oracle Linux.

Assumptions

- You are logged on as the `oracle` user on a `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

1. To verify that the default shell, `bash`, is running, use the `echo` command to display the contents of the `SHELL` variable.

```
[oracle@host01]$ echo $SHELL
/bin/bash
[oracle@host01]$
```

Metacharacter Types	Symbol	Choices or Values
Pathname Expansion	~	Tilde: Represents the home directory of the current user
Expansion	–	Dash: Represents the previous working directory
Parameter Expansion	$	Dollar sign: Parameter/variable expansion
Filename Generation	*	Asterisk: Matches zero or more characters
Filename Generation	?	Question Mark: Matches zero or a single character
Filename Generation	[ ]	Square Brackets: Matches a single character

2. Switch to the user's home directory by using the tilde (~) metacharacter with the `cd` command.

```
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ cd lab/Documents
[oracle@host01]$ pwd
/home/oracle/lab/Documents
[oracle@host01]$ cd ~
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$
```

3. Switch between the `/home/oracle` and `/tmp` directories by using the dash (−) metacharacter with the `cd` command.

```
[oracle@host01]$ cd /tmp
[oracle@host01]$ pwd
/tmp
[oracle@host01]$ cd -
/home/oracle
[oracle@host01]$ cd -
/tmp
[oracle@host01]$ cd
```

Note: There are two shell variables that hold the values for the dash (−) metacharacter. They are `$PWD` and `$OLDPWD`.

4. List all the files and directories in the `~/lab` directory that end with the number 2.

```
[oracle@host01]$ cd ~/lab
[oracle@host01]$ ls *2
file.2   file2    fruit2

dir2:
beans    notes
[oracle@host01]$
```

5. List all the files and directories that start with the string "`file`" and are followed by any other single character.

```
[oracle@host01]$ ls file?
file1   file2   file3   file4
[oracle@host01]$
```

6. List all the files and directories that start with letters `m` through `z` using square brackets.

```
[oracle@host01]$ ls [m-z]*
myvars   tutor.vi

practice:
mailbox  project  projection  research  results

practice1:
appointments  file.1  file.2  play
[oracle@host01]$
```

Practice 5-2: Using Command Redirection

Overview

In this practice, you will perform redirection of standard output (stdout) and standard error (stderr) by using the > (greater than) and | (pipe) metacharacters.

Assumptions

- You are logged on as the oracle user on the host01 system. (See *Course Practice Environment: Security Credentials* for the oracle user password.)
- You have opened a terminal window.

Tasks

1. Use the greater than (>) metacharacter to redirect the list of files and subdirectories of the user's home directory into the dir-list file.

```
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ ls > dir-list
[oracle@host01]$ ls
dir-list    example   Music      records    Videos
celery    Documents  house     Pictures  Templates
Desktop   Downloads  lab        Public     veggies
[oracle@host01]$ cat -n dir-list
     1   celery
     2   Desktop
     3   dir-list
     4   Documents
     5   Downloads
     6   eKit
     7   house
     8   lab
     9   Music
    10   Pictures
    11   Public
    12   records
    13   Templates
    14   veggies
    15   Videos
[oracle@host01]$
```

2. Use the `rm` command to remove the `dir-list` file.

```
[oracle@host01]$ rm dir-list
[oracle@host01]$ ls
Documents  house  Pictures  Templates
celery  Downloads  lab  Public  veggies
Desktop  example  Music  records  Videos
[oracle@host01]$
```

3. From the `/home/oracle` directory, redirect both the standard output (`stdout`) and the standard error (`stderr`) message to a newly created file called `error` in the `lab` directory.

```
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ touch lab/error
[oracle@host01]$ ls /var /test 1> lab/error 2>&1
[oracle@host01]$ cat lab/error
ls: cannot access '/test': No such file or directory
/var:
account
adm
cache


... Output truncated ...
```

4. Use the `rm` command to remove the `lab/error` file.

```
[oracle@host01]$ rm lab/error
[oracle@host01]$ ls lab/error
ls: cannot access 'lab/error': No such file or directory
[oracle@host01]$
```

5. View a list of all the subdirectories located in the `/etc` directory by using the redirection symbol `|` (pipe).

```
[oracle@host01]$ ls -F /etc | grep "/"
abrt/
alsa/
alternatives/
at-spi2/
audisp/
audit/

... Output truncated ...
```

Practice 5-3: Using Variables in the `bash` Shell

Overview

In this practice, you will use variables to store values.

Assumptions

- You are logged on as the oracle user on the host01 system. (See *Course Practice Environment: Security Credentials* for the oracle user password.)
- You have opened a terminal window.

Tasks

1. Use the dollar sign ($) parameter expansion metacharacter to display the value stored inside the SHELL variable by using the echo command.

```
[oracle@host01]$ echo $SHELL
/bin/bash
[oracle@host01]$
```

2. List all shell variables and their values by using the set command.

```
[oracle@host01]$ set
ABRT_DEBUG_LOG=/dev/null
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:expand_aliases:extglob:extquote:fo
rce_fignore:histappend:interactive_comments:login_shell:progcomp
:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=()

... Output truncated ...
```

3. Use the | (pipe) metacharacter and the wc -l word count command to get a count of the number of variables in each environment.

```
[oracle@host01]$ set | wc -l
2728
[oracle@host01]$
```

4. Modify the default prompt by using the PS1 variable.

   a. The default value for PS1 in your activity environment is \u@\h:\W\$. Verify the current prompt setting by using the echo $PS1 command. Then use PS1="$LOGNAME@`uname -n` \$PWD $ " to add the full directory path information to the displayed prompt.

   ```
   [oracle@host01]$ echo $PS1
   [\u@\h \W]$
   [oracle@host01]$ PS1="$LOGNAME`uname -n` \$PWD \$ "
   oracle@host01 /home/oracle $
   ```

   Note: Type the command as it is. The backtick (` `) symbols do not represent single quotation marks. The updated prompt then displays the login name of the user, host name, and the current working directory path. Use the man bash pages and search for *prompting* to see all the special characters that can be used when creating the prompt.

   b. Return the prompt to its default setting.

   ```
   host01 /home/oracle $ PS1="[\u@\h \W]\$ "
   [oracle@host01]$
   ```

   Note: Setting the prompt this way only applies to the current terminal session i.e., the change is not persistent. If you happen to close the current terminal session and open a new terminal session, the original default PS1 value will be used again.

5. If you closed the terminal in the previous task, open a new terminal session by right-clicking the desktop and selecting **Open Terminal**. Then, display the current list of colon (:)-separated values in the PATH variable.

   ```
   [oracle@host01]$ echo $PATH
   /home/oracle/.local/bin:/home/oracle/bin:/home/oracle/.local/bin
   :/home/oracle/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/s
   bin
   [oracle@host01]$
   ```

   Note: The PATH variable is used to store a colon (:)-separated list of directories to be searched when a command is entered.

6. In the user's home directory, add a new directory called sbin. Then append the new directory to the end of the contents in the PATH variable and display the results by using the echo $PATH command.

   ```
   [oracle@host01]$ pwd
   /home/oracle
   [oracle@host01]$ mkdir sbin
   [oracle@host01]$ PATH=$PATH:~/sbin
   [oracle@host01]$ echo $PATH
   /home/oracle/.local/bin:/home/oracle/bin:/home/oracle/.local/bin
   :/home/oracle/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/s
   bin:/home/oracle/sbin
   [oracle@host01]$
   ```

Practice 5-4: Displaying Command History

Overview

In this practice, you will view and set values to manage command-line history.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Before You Begin

Two environment variables control the amount of command-line history that the `bash` shell maintains:

- `HISTFILESIZE`: Controls how many lines of command history are recorded in the `~/.bash_history` file
- `HISTSIZE`: Controls how many command lines are buffered in an open terminal window, which will then be appended to the `~/.bash_history` file when you exit the terminal window

Tasks

1. Check the current number of command lines maintained by the `history` command and set the number of lines being maintained by the `HISTSIZE` variable to `20`.

   ```
   [oracle@host01]$ echo $HISTFILESIZE $HISTSIZE
   1000 1000
   [oracle@host01]$ HISTSIZE=20
   [oracle@host01]$
   ```

2. Confirm that the command-line history for an open terminal window size is set to 20.

   ```
   [oracle@host01]$ echo $HISTSIZE
   20
   [oracle@host01]$
   ```

3. View the page-wise output of the `history` command.

   Note: The following series of `history` command output lines may not match your output based on the actual commands you have entered and that were stored in the `~/.bash_history` file.

```
[oracle@host01]$ history | less
  350  touch lab/error
  351  ls /test
  352  ls /var /test 1> lab/error 2>&1
  353  cat lab/error
  354  rm lab/error
  355  ls lab/error
  356  ls -F /etc | grep "/"
  357  echo $SHELL
  358  echo $PS1
  359  echo $PATH
  360  pwd
  361  mkdir sbin
  362  PATH=$PATH:~/sbin
  363  mkdir sbin
  364  echo $PATH
  365  echo $HISTFILESIZE
  366  echo $HISTFILESIZE $HISTSIZE
  367  HISTSIZE=20
  368  echo $HISTFILESIZE $HISTSIZE
  369  history | less
(END)
```

   Note: Press the **q** key to quit the output from the `less` command.

4. To view the preceding 10 commands from the history database:

```
[oracle@host01]$ history 10
  362  PATH=$PATH:~/sbin
  363  mkdir sbin
  364  echo $PATH
  365  echo $HISTFILESIZE
  366  echo $HISTFILESIZE $HISTSIZE
  367  HISTSIZE=20
  368  echo $HISTFILESIZE $HISTSIZE
  369  history | less
  370  history | less
  371  history 10
[oracle@host01]$
```

5. From the preceding output, to re-execute a specific command from history, enter !368, which repeats command # 368. Choose a relevant command number that appears in your output.

```
[oracle@host01]$ !368
echo $HISTFILESIZE $HISTSIZE
500 20
[oracle@host01]$
```

6. To search the history database, press the Ctrl + R keys at the same time. Then enter the string SIZE all in caps. After the command containing the string is found:

- If this is not the command you were looking for, pressing Ctrl + R continues the search, or if this not the command you choose to execute, then press Ctrl + C to cancel the search

- If this is the command you were searching for, then press the Return/Enter key to execute

```
[oracle@host01]$ <Ctrl+R>
(reverse-i-search)`SIZE': echo $HISTSIZE
```

Note: The search is case-sensitive.

7. Use the -c option to clear previous history.

!!

```
$ history -c
$ history
 354 history
$
```

8. The following are the various methods for repeating the previous command quickly:

- Use the up arrow to view the previous command and press Enter to execute it.
- Enter !! and press Enter from the command line.
- Enter !-1 and press Enter from the command line.

Practice 5-5: Customizing the User's Work Environment

Overview

In this practice, you will use the `~/.bashrc` file to make customized changes to your shell environment. Oracle Linux provides a `~/.bashrc` file as part of the `bash` shell configuration.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

1. Confirm that you are currently in the home directory by using the `pwd` command and then display the contents of the `.bashrc` file.

```
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-
paging feature:
# export SYSTEMD_PAGER=

# User specific alias and functions
[oracle@host01]$
```

2. In the previous practice, the shell variable `HISTSIZE`, which controls how many command lines are buffered in an open terminal window, was set to 20.

   To make this change permanent, you add the line `HISTSIZE=20` to the end of the `.bashrc` file.

   Note: Any changes that you want to make permanently to the `bash` shell environment can be added to the end of the `~/.bashrc` file.

a. Use the `vi .bashrc` command to edit the `.bashrc` file. Use the **G** (capital letter G) key command to go to the bottom of the file, and then press the **o** key to open a new line below the line with the cursor.

```
[oracle@host01]$ vi .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-
paging feature:
# export SYSTEMD_PAGER=

# User specific alias and functions

~
~
~
~
-- INSERT --
```

b. Enter `HISTSIZE=20`

```
[oracle@host01]$ vi .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-
paging feature:
# export SYSTEMD_PAGER=

# User specific alias and functions
HISTSIZE=20
~
~
… OUTPUT TRUNCATED …
```

c. Press the **ESC** key. Then use the :wq command and press **Enter** to save the change and exit.

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-
paging feature:
# export SYSTEMD_PAGER=

# User specific alias and functions
HISTSIZE=20

:wq
[oracle@host01]$
```

3. View the .bashrc file to verify that the command has been added and saved.

```
[oracle@host01]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-
paging feature:
# export SYSTEMD_PAGER=

# User specific alias and functions
HISTSIZE=20
[oracle@host01]$
```

Practices for Lesson 6: Using Basic File Permissions

Practices for Lesson 6: Overview

Practices Overview

In these practices, you will:

- Change file ownership
- Change file permissions
- Modify default permissions

Practice 6-1: Changing File Ownership

Overview

In this practice, you will view and change file ownership.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You opened a terminal window.

Tasks

1. To find the owner of the existing `Documents` directory within the `lab` directory, use the `ls -ld` command. Ensure that you are in the `$HOME` directory.

```
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ ls -ld lab/Documents
drwxrwxr-x. 2 oracle oracle 4096 Nov 26 06:23 lab/Documents
[oracle@host01]$
```

For additional details about the output from the `ls` command, refer to the following table:

Symbol	Meaning	Comments
d	Directory	If it begins with a dash −, it means it is a regular file.
rwx	read, write, and execute	The user's privilege set
r-x	read, write not permitted, and execute	The group's privilege set
r-x	read, write not permitted, and execute	The other's privilege set
2	number of links	
oracle	user/owner	The user who owns the directory/file
oracle	Group	The group which owns the directory/file
38	Size	The size of the file or directory in bytes

Mar 5 17:36	Day, Month, Year and Hours:Minutes	Last modified date/time information
lab/Documents	directory/file name	

2. Identify the owner of the contents in the `Documents` directory by using the `ls -l` command.

```
[oracle@host01]$ ls -l lab/Documents
total 8
-rw-rw-r--. 1 oracle oracle 21 Mar  5  2018 misc.txt
-rw-rw-r--. 1 oracle oracle 28 Mar  5  2018 sample.txt
[oracle@host01]$
```

Note: Observe that `oracle` is not only the owner of the `Documents` directory, but also the owner of the contents of the `Documents` directory.

3. Change the ownership of the `Documents` directory to the `root` user.

Use `su -` to switch to the `root` user and run the change owner, `chown oracle ~oracle/lab/Documents` command. (See *Course Practice Environment: Security Credentials* for the `root` user password)

```
[oracle@host01]$ su -
Password: <Password>
[root@host01]# chown root ~oracle/lab/Documents
[root@host01]# ls -ld ~oracle/lab/Documents
drwxrwxr-x. 2 root oracle 4096 Nov 26 06:23
/home/oracle/lab/Documents
[root@host01]#
```

4. Confirm the ownership of the contents of the `Documents` directory.

Change both `user` and `group` ownership of its contents to `root` and `root` by running the `chown` command again with the recursive `-R` option.

```
[root@host01]# ls -l ~oracle/lab/Documents
total 8
-rw-rw-r--. 1 oracle oracle 21 Mar  5  2018 misc.txt
-rw-rw-r--. 1 oracle oracle 28 Mar  5  2018 sample.txt
[root@host01]# chown -R root:root ~oracle/lab/Documents/
[root@host01]# ls -l ~oracle/lab/Documents
total 8
-rw-rw-r--. 1 root root 21 Mar  5  2018 misc.txt
-rw-rw-r--. 1 root root 28 Mar  5  2018 sample.txt
[root@host01]# exit
logout
[oracle@host01]$
```

Practice 6-2: Changing File Permissions

Overview

In this practice, you will view and change permissions on files.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You opened a terminal window.

Before You Begin

- The `umask` value is set to `0022` on your system. To verify, run the `umask` command.

```
[oracle@host01]$ umask
0002
[oracle@host01]$
```

- If the `umask` is not set to 0022, then set the `umask` value to 0022 by running the following command:

```
[oracle@host01]$ umask 0022
[oracle@host01]$
```

Tasks

1. Create a new directory called `perm` in your `/home/oracle/lab` directory.

```
[oracle@host01]$ cd ~/lab
[oracle@host01]$ mkdir perm
```

2. Change to the `/etc` directory and list these four files – `group`, `motd`, `shadow`, `fstab`.

   Note: For Oracle Linux, there are no permissions on the shadow file.

```
[oracle@host01]$ cd /etc
[oracle@host01]$ ls -l group motd shadow fstab
-rw-r--r--. 1 root root  597 Aug 31 20:00 fstab
-rw-r--r--. 1 root root  956 Sep  3 20:21 group
-rw-r--r--. 1 root root    0 Sep 10  2018 motd
----------. 1 root root 1433 Nov 26 08:50 shadow
[oracle@host01]$
```

3. Copy the four files to your `~/lab/perm` directory. The `shadow` file will fail to copy.

```
[oracle@host01]$ cp group motd shadow fstab ~/lab/perm
cp: cannot open 'shadow' for reading: Permission denied
[oracle@host01]$
```

For Instructor Use Only.

This document should not be distributed.

Copyright © 2022, Oracle and/or its affiliates

4. Go to your `lab` directory and verify the contents of its `~/lab/perm` directory. Copy the contents of the `/etc/skel` directory into the `~/lab/perm` directory.

```
[oracle@host01]$ cd ~/lab
[oracle@host01]$ ls -l perm
total 8
-rw-r--r--. 1 oracle oracle 513 Mar 14 18:39 fstab
-rw-r--r--. 1 oracle oracle 972 Mar 14 18:39 group
-rw-r--r--. 1 oracle oracle   0 Mar 14 18:39 motd
[oracle@host01]$ cp -r /etc/skel perm
[oracle@host01]$
```

5. List the contents of the `perm` directory.

```
[oracle@host01]$ ls -l perm
total 8
-rw-r--r--. 1 oracle oracle 513 Mar 14 18:39 fstab
-rw-r--r--. 1 oracle oracle 972 Mar 14 18:39 group
-rw-r--r--. 1 oracle oracle   0 Mar 14 18:39 motd
drwxr-xr-x. 3 oracle oracle  74 Mar 14 18:44 skel
[oracle@host01]$
```

In the following table, enter the permission sets for each file and write the three-digit octal number that represents the combined set of permissions.

File or Directory	Permission Sets			Octal Value
	User/Owner	Group	Other	
group	rw-	r--	r--	644
motd	rw-	r--	r--	644
skel	Rwx	r-x	r-x	755
vfstab/fstab	rw-	r--	r--	644

The vertical text on right side.

6. Create a new file `test1` and a new directory `test`.

```
[oracle@host01]$ pwd
/home/oracle/lab
[oracle@host01]$ touch test1
[oracle@host01]$ mkdir test
```

a. Examine the default permissions of the new file.

```
[oracle@host01]$ ls -l test1
-rw-r--r--. 1 oracle oracle 0 Mar 14 18:46 test1
[oracle@host01]$
```

b. Check the default permissions of the new directory.

```
[oracle@host01]$ ls -ld test
drwxr-xr-x. 2 oracle oracle 4096 Mar 14 18:47 test
[oracle@host01]$
```

7. Using the `chmod` command and symbolic mode, add write (`w`) permission for the group permission set to the `motd` file.

Note: Symbolic mode uses a combination of letters and symbols to add or remove permissions for each type of user.

```
[oracle@host01]$ chmod g+w perm/motd
[oracle@host01]$ ls -l perm
total 8
-rw-r--r--. 1 oracle oracle 513 Mar 14 18:39 fstab
-rw-r--r--. 1 oracle oracle 972 Mar 14 18:39 group
-rw-rw-r--. 1 oracle oracle   0 Mar 14 18:39 motd
drwxr-xr-x. 3 oracle oracle  74 Mar 14 18:44 skel
[oracle@host01]$
```

8. Using octal mode, change the permissions on the `motd` file to `-rwxrw----`.

Note: Octal mode uses octal numbers to represent permissions. Octal mode is also referred to as absolute mode.

```
[oracle@host01]$ chmod 760 perm/motd
[oracle@host01]$ ls -l perm
total 8
-rw-r--r--. 1 oracle oracle 513 Mar 14 18:39 fstab
-rw-r--r--. 1 oracle oracle 972 Mar 14 18:39 group
-rwxrw----. 1 oracle oracle   0 Mar 14 18:39 motd
drwxr-xr-x. 3 oracle oracle  74 Mar 14 18:44 skel
[oracle@host01]$
```

9. Using octal mode, add write (w) permission for other on the file named group.

```
[oracle@host01]$ chmod 646 perm/group
[oracle@host01]$ ls -l perm
total 8
-rw-r--r--. 1 oracle oracle 513 Mar 14 18:39 fstab
-rw-r--rw-. 1 oracle oracle 972 Mar 14 18:39 group
-rwxrw----. 1 oracle oracle   0 Mar 14 18:39 motd
drwxr-xr-x. 3 oracle oracle  74 Mar 14 18:44 skel
[oracle@host01]$
```

10. Identify the GID and UID for the motd file.

```
[oracle@host01]$ ls -n perm/motd
-rwxrw----. 1 1000 1000 0 Mar 14 18:39 perm/motd
[oracle@host01]$
```

11. Create a new file called memo in your dir4 directory.

```
[oracle@host01]$ touch ~/lab/dir4/memo
[oracle@host01]$ ls -l ~/lab/dir4/memo
-rw-r--r--. 1 oracle oracle 0 Mar 14 18:54
/home/oracle/lab/dir4/memo
[oracle@host01]$
```

12. Remove the read (r) permission for the owner from the memo file in the dir4 directory. You can use symbolic mode to do this.

```
[oracle@host01]$ chmod u-r ~/lab/dir4/memo
[oracle@host01]$ ls -l ~/lab/dir4/memo
--w-r--r--. 1 oracle oracle 0 Mar 14 18:54
/home/oracle/lab/dir4/memo
[oracle@host01]$
```

Or you can use octal mode.

```
[oracle@host01]$ chmod 244 ~/lab/dir4/memo
[oracle@host01]$ ls -l ~/lab/dir4/memo
--w-r--r--. 1 oracle oracle 0 Mar 14 18:54
/home/oracle/lab/dir4/memo
[oracle@host01]$
```

13. Use the `cat` command to view the `memo` file.

```
[oracle@host01]$ cat ~/lab/dir4/memo
cat: /home/oracle/lab/dir4/memo: Permission denied
[oracle@host01]$
```

Note: This fails because read permission has been removed from the user. Even though you are part of the group, the permissions are viewed in the order in which they appear.

14. Copy the `memo` file to the `~/lab` directory.

```
[oracle@host01]$ cp ~/lab/dir4/memo ~/lab
cp: cannot open '/home/oracle/lab/dir4/memo' for reading:
Permission denied
[oracle@host01]$
```

Note: You cannot copy the file because the user has no read permission.

Practice 6-3: Modifying Default Permissions

Overview

In this practice, you will modify the default permissions of files and directories.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You opened a terminal window.

Tasks

1. Check the current `umask` value on your system with the `umask` command.

   Note: The `umask` utility modifies the default permissions set for files and directories at the time of creation.

   ```
   [oracle@host01]$ umask
   0022
   [oracle@host01]$
   ```

2. Change `umask` to 027.

   ```
   [oracle@host01]$ umask 027
   [oracle@host01]$
   ```

3. Create a new file and a new directory in the `lab` directory. Record the access permissions.

   ```
   [oracle@host01]$ cd
   [oracle@host01]$ touch lab/testfile
   [oracle@host01]$ mkdir lab/testdir
   [oracle@host01]$ ls -l lab/testfile
   -rw-r-----. 1 oracle oracle 0 Mar 14 19:27 lab/testfile
   [oracle@host01]$ ls -ld lab/testdir
   drwxr-x---. 2 oracle oracle 6 Mar 14 19:28 lab/testdir
   [oracle@host01]$
   ```

4. Change `umask` back to `0022`.

   ```
   [oracle@host01]$ umask 0022
   ```

5. Create a new file and a new directory.

   ```
   [oracle@host01]$ touch lab/test2file
   [oracle@host01]$ mkdir lab/test2dir
   ```

6. Record the access permissions.

```
[oracle@host01]$ ls -l lab/test2file
-rw-r--r--. 1 oracle oracle 0 Mar 14 19:29 lab/test2file
[oracle@host01]$ ls -ld lab/test2dir
drwxr-xr-x. 2 oracle oracle 4096 Mar 14 19:29 lab/test2dir
[oracle@host01]$
```

Note: The permission set for other's using 0027 has no privileges, whereas with 0022, the permission set for other's has read access on files, and read and execute access on directories.

Practices for Lesson 6: Using Basic File Permissions

Practices for Lesson 7: Performing Basic Process Control

Practices for Lesson 7: Overview

Practices Overview

In this practice, you will:

- List system processes
- Control system processes
- Terminate a process

Practices for Lesson 7: Performing Basic Process Control

Practice 7-1: Controlling System Processes

Overview

In this practice, you will determine the process identifier (PID), view a process tree, and kill processes.

Before You Begin

This practice introduces the `tty` command, which displays the name of the current terminal window. The name displayed by the `tty` command includes a unique identification number assigned by the Linux operating system to each open terminal window (for example, `/dev/pts/2`). In the tasks illustrating the `tty` command, the unique identification number is displayed as `/dev/pts/`*n*, where *n* is a numeral.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You opened a terminal window.

Tasks

1. Use the following `ps` commands to list the processes currently running on your system.

   ```
   [oracle@host01]$ ps
     PID TTY            TIME CMD
   21786 pts/0      00:00:00 bash
    6443 pts/0      00:00:00 ps
   [oracle@host01]$
   ```

   Note: This command prints information for the current user and terminal.

2. Use the `-f` option to print a full listing for the command.

   ```
   [oracle@host01]$ ps -f
   UID        PID  PPID  C STIME TTY          TIME CMD
   oracle  21786 21781  0 Mar12 pts/0    00:00:00 bash
   oracle   6461 21786  0 17:35 pts/0    00:00:00 ps -f
   [oracle@host01]$
   ```

3. Use the `-e` option to print information about every process running. Then use the `ps -e | wc -l` command to show the total number of processes.

```
[oracle@host01]$ ps -e
  PID TTY              TIME CMD
    1 ?            00:00:57 systemd
    2 ?            00:00:00 kthreadd
    3 ?            00:00:00 ksoftirqd/0
    6 ?            00:00:04 kworker/30:0
    7 ?            00:00:16 rcu_sched
    8 ?            00:00:00 rcu_bh
    9 ?            00:00:15 rcuos/0
   10 ?            00:00:00 rcuob/0
   11 ?            00:00:00 migration/0

... Output truncated ...

[oracle@host01]$ ps -e | wc -l
179
[oracle@host01]$
```

4. Run the `ps -f` command again.

   Note: Observe the TTY column where the controlling terminal is `pts/0`

```
[oracle@host01]$ ps -f
UID        PID  PPID  C STIME TTY          TIME CMD
oracle   21786 21781  0 Mar12 pts/0    00:00:00 bash
oracle    6461 21786  0 17:37 pts/0    00:00:00 ps -f
[oracle@host01]$
```

5. Open a second terminal window and execute the `ps -f` command in the new terminal window.

   Note: Observe the TTY column in the VM, where the controlling terminal is `pts/1`

   2nd terminal window:

```
[oracle@host01]$ ps -f
UID       PID  PPID  C STIME TTY          TIME CMD
oracle   6514 21781  0 17:38 pts/1    00:00:00 bash
oracle   6553  6514  0 04:05 pts/1    00:00:00 ps -f
[oracle@host01]$
```

6. In your first terminal window, enter the gnome-calculator command:

   1ˢᵗ terminal window:

   ```
   [oracle@host01]$ gnome-calculator
   ```

   The Gnome Calculator tool opens.

7. In the second terminal window, use the ps -ef | grep gnome-calculator or pgrep
   -f gnome-calculator commands to identify the PID of the gnome-calculator
   process.

   Note: As you launched Gnome Calculator from the first terminal window, command line
   input is not available in this terminal while the utility is open.

   2ⁿᵈ terminal window:

   ```
   [oracle@host01]$ ps -ef | grep gnome-calculator
   oracle  6590    21786 0  17:40 pts/0    00:00:00 gnome-calculator
   oracle  6598     6514 0  17:40 pts/1    00:00:00 grep -color=auto
   gnome-calculator
   [oracle@host01]$ pgrep -f gnome-calculator
   6590
   [oracle@host01]$
   ```

   Note: In the preceding output, 6590 is the PID value for the gnome-calculator. Your
   PID value will be different.

8. From the second terminal window, use the kill <*PID*> command or the pkill -f
   gnome-calculator command to terminate the gnome-calculator process. The -f
   option used with pkill ensures the process name fully matches the name used in the
   command.

   2ⁿᵈ terminal window:

   ```
   [oracle@host01]$ kill 6590
   ```

   Or use:

   ```
   [oracle@host01]$ pkill -f gnome-calculator
   ```

9. In the second terminal window, enter the `tty` command to identify the name of this terminal window. The name appears as `/dev/pts/<n>`, where *n* is a number (for example, `/dev/pts/4`).

    2nd terminal window:

    ```
    [oracle@host01]$ tty
    /dev/pts/1
    [oracle@host01]$
    ```

10. Return to your first terminal window. Use the `pgrep -t` (terminal option) command to find the PID associated with the second terminal window.

    1st terminal window:

    ```
    [oracle@host01]$ pgrep -t pts/1
    6514
    [oracle@host01]$
    ```

    Note: Your PID value will be different.

11. In your first terminal window, use the `kill` command or the `pkill -t` command attempt to terminate your second terminal window.

    1st terminal window:

    ```
    [oracle@host01]$ kill 6514
    [oracle@host01]$
    ```

    Or use:

    ```
    [oracle@host01]$ pkill -t pts/1
    [oracle@host01]$
    ```

12. [Optional] You can also use the `kill` command or the `pkill` command with the `-9` option to terminate your second terminal window.

    1st terminal window:

    ```
    [oracle@host01]$ kill -9 6514
    [oracle@host01]$
    ```

    Or use:

    ```
    [oracle@host01]$ pkill -9 -t pts/1
    [oracle@host01]$
    ```

    Note: The `-9` option forces the terminal process to terminate.

13. Run the following `kill -l` (list option) commands to identify the signal names and signal values.

```
[oracle@host01]$ kill -l 9
KILL
[oracle@host01]$ kill -l kill
9
[oracle@host01]$ kill -l 15
TERM
[oracle@host01]$ kill -l term
15
[oracle@host01]$
```

Note: For signal value 9, the signal name is KILL, and for the signal name kill, the signal value is 9. For signal value 15, the signal name is TERM, and for the signal name term, the signal value is 15.

14. In the terminal window, enter the `sleep 600 &` command and place it in the background.

```
[oracle@host01]$ sleep 600 &
[1] 6877
[oracle@host01]$
```

15. Use the `ps` command to identify the bash shell process running in that window.

```
[oracle@host01]$ ps
  PID TTY          TIME CMD
21786 pts/0     00:00:00 bash
 6877 pts/0     00:00:00 sleep
 6881 pts/0     00:00:00 ps
[oracle@host01]$
```

Note: Your PID value will be different.

16. Open a second terminal window. To display the process tree, use the `pstree -p` command along with the bash shell PID.

    2nd terminal window:

```
[oracle@host01]$ pstree -p 1252
bash(21786)---sleep(6877)
[oracle@host01]$
```

17. In the second terminal window, terminate the first terminal window by using the `kill -9` command with the bash shell PID.

    2nd terminal window:

```
[oracle@host01]$ kill -9 21786
```

Practices for Lesson 7: Performing Basic Process Control

Practices for Lesson 8: Using Advanced Shell Features in Shell Scripts

Practices for Lesson 8: Overview

Practices Overview

In this practice, you will:

- Use the advanced bash shell functionality

- Use shell scripts

Practice 8-1: Using the Advanced Bash Shell Functionality

Overview

In this practice, you will perform some tasks using the job control commands.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You opened a terminal window.

Tasks

1. Run the `sleep 500 &` command to create a running job.

   ```
   [oracle@host01]$ sleep 500 &
   [1] 13462
   [oracle@host01]$
   ```

2. Job control commands enable you to place jobs in the foreground or background, and to start or stop jobs. Use the `jobs` command to confirm the `sleep` command executed is currently running.

   ```
   [oracle@host01]$ jobs
   [1]+  Running                 sleep 500 &
   [oracle@host01]$
   ```

3. Bring the job to the foreground, and then put it back in the background. To stop a command and get back to the prompt, use `CTRL+Z`.

   ```
   [oracle@host01]$ fg %1
   sleep 500
   ^Z
   [1]+  Stopped                 sleep 500
   [oracle@host01]$ bg %1
   [1]+ sleep 500 &
   [oracle@host01]$
   ```

   Note: The `jobs` command lists all jobs that are currently running or are stopped in the background. The `bg %n` command runs the current or specified job in the background (`n` is the job ID).

4. Terminate a job with the `kill` command. To confirm, run the `jobs` command again.

   ```
   [oracle@ol7-server1 ~]$ kill %1
   [1]+  Terminated              sleep 500
   [oracle@ol7-server1 ~]$ jobs
   [oracle@ol7-server1 ~]$
   ```

5. Enable the `noclobber` option and use the `set` command to verify that it is enabled.

```
[oracle@host01]$ set -o noclobber
[oracle@host01]$ set -o | more
allexport       off
braceexpand     on
emacs           on
errexit         off
errtrace        off
functrace       off
hashall         on
histexpand      on
history         on
ignoreeof       off
interactive-comments on
keyword         off
monitor         on
noclobber           on
noexec          off
noglob          off
nolog           off
notify          off
nounset         off
onecmd          off
physical        off
pipefail        off
posix               off
privileged          off
verbose             off
vi                  off
xtrace              off
[oracle@host01]$
```

Note: To stop the command output, use `ctrl` + `z` or `q`.

6. Display all predefined aliases.

```
[oracle@ol7-server1 ~]$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi=vim
alias which='alias | /usr/bin/which --tty-only -read-alias --
show-dot --show-tilde'
[oracle@ol7-server1 ~]$
```

Note: If you do not see an alias list, there are no predefined aliases in the system.

7. Create an alias named `cls` that clears the terminal screen.

```
[oracle@host01]$ alias cls=clear
```

8. Create an alias named `dir` that displays a long listing of all the files and directories in the current directory.

```
[oracle@host01]$ alias dir='ls -l'
```

9. Create an alias named `h` that lists your command `history`.

```
[oracle@host01]$ alias h=history
```

10. Run the `alias` command again.

```
[oracle@host01]$ alias
alias cls=clear
alias dir='ls -l'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias h=history
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi=vim
alias which='alias | /usr/bin/which --tty-only -read-alias --
show-dot --show-tilde'
[oracle@host01]$
```

11. Unalias the `history` command and the `cls` command.

```
[oracle@host01]$ unalias h
[oracle@host01]$ unalias cls
```

12. Display all defined functions.

```
[oracle@host01]$ typeset -f
__expand_tilde_by_ref ()
{
    if [[ ${!1} == \~* ]]; then
        if [[ ${!1} == */* ]]; then
            eval $1="${!1/%\/*}"/'${!1#*/}';
        else
            eval $1="${!1}";
        fi;
    fi
}
__get_cword_at_cursor_by_ref ()
{

... Output truncated ...

quote_readline ()
{
    local quoted;
    _quote_readline_by_ref "$1" ret;
    printf %s "$ret"
}
[oracle@host01]$
```

Note: There is a lot of information returned for this command in Oracle Linux. Only the beginning and end of the output is shown above.

13. Create and test a function called data that performs the following activities:

- Clears the terminal screen
- Displays the date and time
- Displays users logged in to the system
- Displays the path of the current working directory
- Lists the current working directory in a long format

```
$ function data { clear; date; who; pwd; ls -l; }
$
```

Practices for Lesson 8: Using Advanced Shell Features in Shell Scripts

14. To confirm that the function is created, run `typeset -f data`.

```
[oracle@host01]$ typeset -f data
data ()
{
    clear;
    date;
    who;
    pwd;
    ls --color=auto -l
}
[oracle@host01]$
```

15. Use `vi` to edit the `.bash_profile` file in your home directory.

```
[oracle@host01]$ vi ~/.bash_profile
```

   a. Add the following line entries to the profile file:

```
set -o vi
alias h='history'
alias cls='clear'
alias lf='pwd; ls -lF'
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

set -o vi
alias h='history'
alias cls='clear'
alias lf='pwd; ls -lF'
~
~
```

16. Log your user out and back in again, and then test your new aliases and functions with the commands `h`, `lf`, and `clr`. Verify that the output is as expected. When you finish testing, close the terminal with the `exit` command.

Practice 8-2: Using Shell Scripts

Overview

In this practice, you will edit and run shell scripts using some of the test and conditional statements.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You opened a terminal window.

Tasks

1. In this task, you will modify and run a simple shell script called `info.sh`, which displays the date, time, username, and current directory.

    a. Switch to the `lab` directory in your home directory. Open the `vi` editor and edit the shell script file, `info.sh`, so it is as shown below. Ensure that you are in the `lab` directory of your home directory.

    Note: For your benefit, the `info.sh` file is already available in the `lab` directory.

    ```
    [oracle@host01]$ cd lab
    [oracle@host01]$ vi info.sh
    #!/usr/bin/bash
    #info.sh
    # This script displays the date, time, username and the current
    directory.
        echo "Date and time is:"
        date
        echo
        echo "Your username is: `whoami`"
        echo "Your current directory is: `pwd`"
    ~
    ~
    ```

    Note: Exit the file by pressing the ESC key followed by executing the `:q` command to quit the `vi` editor. The first entry `#!/usr/bin/bash` indicates that the script should be run in the bash shell.

Practices for Lesson 8: Using Advanced Shell Features in Shell Scripts

b. Grant execute permission to the script by running the `chmod +x` command. Confirm this change by running the `ls -l` command.

```
[oracle@host01]$ chmod +x info.sh
[oracle@host01]$ ls -l info.sh
-rwxr-xr-x. 1 oracle oracle 232 May 13  2017 info.sh
[oracle@host01]$
```

c. To execute the script, run the command as shown below:

```
[oracle@host01]$ ./info.sh
Date and time is:
Tue Feb 27 14:59:44 IST 2018


Your username is: oracle
Your current directory is: /home/oracle/lab
[oracle@host01]$
```

2. In this task, you pass values to the `greetings` shell script, which is also available in the `lab` directory.

a. View greetings by running the `cat` command.

```
[oracle@host01]$ cat greetings
#!/bin/sh
echo $1 $2 #echo the first two parameters passed
[oracle@host01]$
```

b. Add user execute permissions to `greetings`.

```
[oracle@host01]$ chmod u+x greetings
[oracle@host01]$
```

c. Run greetings with the `hello` and `world` values.

```
[oracle@host01]$ ./greetings hello world
hello world
[oracle@host01]$
```

3. In this task, you will practice using the `test` command.

a. Test whether the value of the `LOGNAME` variable is student.

```
[oracle@host01]$ echo $LOGNAME
oracle
[oracle@host01]$ test "$LOGNAME" = "oracle"
[oracle@host01]$ echo $?
0
[oracle@host01]$
```

Note: The test command does not return any output. For a true condition, the exit status of the test command is set to `0`.

b. Now test whether the value of the LOGNAME variable is user.

```
[oracle@host01]$ test "$LOGNAME" = "user"
[oracle@host01]$ echo $?
1
[oracle@host01]$
```

4. In this task, you will practice using conditional statements.

a. Use cat to view the shell script called leaptest.sh, which is provided in your
    /home/oracle/lab directory.

```
[oracle@host01]$ cat leaptest.sh
#!/usr/bin/bash
# This script will test if the year is a leap year.


year=`date +%Y`


if [ $[$year % 400] -eq "0" ]; then
  echo "This is a leap year.  February has 29 days."
elif [ $[$year % 4] -eq 0 ]; then
        if [ $[$year % 100] -ne 0 ]; then
          echo "This is a leap year, February has 29 days."
        else
          echo "This is not a leap year.  February has 28 days."
        fi
else
  echo "This is not a leap year.  February has 28 days."
fi


[oracle@host01]$
```

b. Add execute permission to the script.

```
[oracle@host01]$ chmod u+x leaptest.sh
```

c. Find the current year using the date command and then subsequently run the
    leaptest script to find whether the current year is a leap year.

```
[oracle@host01]$ date
Fri Nov 27 10:01:47 UTC 2020
[oracle@host01]$ ./leaptest.sh
This is a leap year, February has 29 days.
[oracle@host01]$
```

Practices for Lesson 8: Using Advanced Shell Features in Shell Scripts

Practices for Lesson 9: Archiving, Compressing, and Performing Remote File Transfers

Practices for Lesson 9: Overview

Practices Overview

In these practices, you will:

- Archive and retrieve files

- Compress and restore files

- Establish remote connections and file transfers

Practice 9-1: Archiving and Retrieving Files

Overview

In this practice, you will perform tasks to create, view, and retrieve archived files.

Note

- You will perform the practices in your `/home/oracle` directory. If you are in a different directory when starting, use the `cd` command to change the directory to the `/home/oracle` directory.
- Bash is the default shell.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

1. Run `chmod 775` on the `/home/oracle/lab` directory before starting this practice.

```
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ chmod -R 775 lab
[oracle@host01]$
```

2. Archive the `lab` directory in your home directory to a file called `lab.tar` by using the `tar` command.

```
[oracle@host01]$ pwd
/home/oracle
```

```
[oracle@host01]$ ls -l
total 8
-rw-rw-r--.  1 oracle oracle   51 Mar 13 18:28 £1
-rw-rw-r--.  1 oracle oracle    0 Mar 13 14:31 celery
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Desktop
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Documents
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Downloads
-rw-rw-r--.  1 oracle oracle   53 Mar 13 17:18 example
drwxrwxr-x.  2 oracle oracle    6 Mar 13 14:33 house
drwxrwxr-x. 13 oracle oracle 4096 Mar 19 17:15 lab
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Music
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Pictures
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Public
drwxrwxr-x.  2 oracle oracle    6 Mar 13 14:31 records
drwxrwxr-x.  2 oracle oracle    6 Mar 14 14:46 sbin
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Templates
drwxrwxr-x.  2 oracle oracle    6 Mar 13 14:34 veggies
drwxr-xr-x.  2 oracle oracle    6 Mar 13  2017 Videos

[oracle@host01]$ tar cvf lab.tar lab
lab/
lab/fruit2
lab/file.1
lab/leaptest.sh
lab/Documents/
lab/Documents/misc.txt
lab/Documents/sample.txt

... Output truncated ...

lab/testfile
lab/testdir/
lab/test2file
lab/test2dir/
lab/feathers
lab/info.sh
```

```
[oracle@host01]$ ls -l
total 128
-rw-rw-r--.  1 oracle oracle     51 Mar 13 18:28 £1
-rw-rw-r--.  1 oracle oracle      0 Mar 13 14:31 celery
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Desktop
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Documents
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Downloads
-rw-rw-r--.  1 oracle oracle     53 Mar 13 17:18 example
drwxrwxr-x.  2 oracle oracle      6 Mar 13 14:33 house
drwxrwxr-x. 13 oracle oracle   4096 Mar 19 17:15 lab
-rw-rw-r--.  1 oracle oracle 112640 Mar 19 17:46 lab.tar
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Music
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Pictures
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Public
drwxrwxr-x.  2 oracle oracle      6 Mar 13 14:31 records
drwxrwxr-x.  2 oracle oracle      6 Mar 14 14:46 sbin
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Templates
drwxrwxr-x.  2 oracle oracle      6 Mar 13 14:34 veggies
drwxr-xr-x.  2 oracle oracle      6 Mar 13  2017 Videos
[oracle@host01]$
```

Note: You can follow similar steps to create and archive the file to another directory or an external drive.

3. Create a new directory called `retrieve` under the `lab` directory in your home directory. Use the `cd` command to move to this new directory. Use this new directory to practice retrieving files from archives. Retrieve the contents of the `lab.tar` file that you just created into this new directory.

```
[oracle@host01]$ cd lab
[oracle@host01]$ mkdir retrieve
[oracle@host01]$ cd retrieve
[oracle@host01]$ tar xvf /home/oracle/lab.tar
lab/
lab/fruit2
lab/file.1
lab/leaptest.sh
lab/Documents/
lab/Documents/misc.txt
lab/Documents/sample.txt

... Output truncated ...
```

```
lab/testfile
lab/testdir/
lab/test2file
lab/test2dir/
lab/feathers
lab/info.sh
[oracle@host01]$ ls -l
total 4
drwxrwxr-x. 13 oracle oracle 4096 Mar 19 17:15 lab
[oracle@host01]$ cd
[oracle@host01]$
```

Practice 9-2: Compressing and Restoring Files

Overview

In this practice, you will compress, view, and uncompress files.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Tasks

1. In the `/home/oracle/lab` directory, compress the `dante` and `file1` files by using the `gzip` command.

```
[oracle@host01]$ cd
[oracle@host01]$ pwd
/home/oracle
[oracle@host01]$ cd lab
[oracle@host01]$ gzip dante
[oracle@host01]$ gzip file1
[oracle@host01]$ ls -l dante* file1*
-rw-r--r--. 1 oracle oracle 368 Mar  5 17:36 dante_1
-rw-r--r--. 1 oracle oracle 768 Mar  5 17:36 dante.gz
-rw-r--r--. 1 oracle oracle 863 Mar  5 17:36 file1.gz
[oracle@host01]$
```

Note: When using the `gzip` command, the output file extension will be `.gz`. The new names for compressed versions are `dante.gz` and `file1.gz`.

In addition, you can use the following `zcat` command to view the contents of the file that was compressed with the `gzip` command: `zcat <filename>`

```
[oracle@host01]$ zcat dante.gz
                        The Life and Times of Dante


                           by Dante Pocai



Mention "Alighieri" and few may know about whom you are talking.
Say "Dante," instead, and the whole world knows whom you mean.
For Dante Alighieri, like Raphael, Michelangelo, Galileo, etc.
is usually referred to by his first name.  There is only one
Dante, as we recognize one Raphael, one Michelangelo, and one
Galileo.


... Output truncated ...


[oracle@host01]$
```

Note: Including the file extension `.gz` is optional for the `zcat` command.

2. Use the `gunzip` command to uncompress the `dante.gz` and `file1.gz` files.

```
[oracle@host01]$ gunzip dante file1
[oracle@host01]$ ls -l dante* file1*
-rw-r--r--. 1 oracle oracle 1319 Mar  5 17:36 dante
-rw-r--r--. 1 oracle oracle  368 Mar  5 17:36 dante_1
-rw-r--r--. 1 oracle oracle 1610 Mar  5 17:36 file1
[oracle@host01]$
```

Note: Including the file extension `.gz` is optional for the `gunzip` command.

3. Use the zip command to archive and compress the `file3`, `fruit2`, and `tutor.vi` files to a single file called `myfiles.zip`.

Note: The original versions of the `file3`, `fruit2`, and `tutor.vi` files still exist after archiving and compression.

```
[oracle@host01]$ zip myfiles.zip file3 fruit2 tutor.vi
  adding: file3 (deflated 26%)
  adding: fruit2 (deflated 14%)
  adding: tutor.vi (deflated 74%)
[oracle@host01]$ ls -l myfiles.zip
-rw-rw-r--. 1 oracle oracle 7983 Mar 19 18:00 myfiles.zip
[oracle@host01]$
```

4. Use the `unzip -l` command to view the newly compressed archive file.

```
[oracle@host01]$ unzip -l myfiles.zip
Archive:  myfiles.zip
  Length      Date     Time    Name
---------  ---------- -----    ----
      218  05-13-2017 22:59    file3
       57  05-13-2017 22:59    fruit2
    28709  05-13-2017 22:59    tutor.vi
---------                      -------
    28984                      3 files
[oracle@host01]$
```

5. Use the `unzip` command to extract and uncompress the files in the `myfiles.zip` archive.

```
[oracle@host01]$ unzip myfiles.zip
Archive: myfiles.zip
replace file3? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
inflating: file3
inflating: fruit2
inflating: tutor.vi
[oracle@host01]$
```

Note: The options [y], [n], [A], and [N] allow you to control how the archived files are extracted.

Practice 9-3: Establishing Remote Connections and File Transfers

Overview

In this practice, you will use remote connection and file transfer commands.

Assumptions

- You are logged on as the `oracle` user on the `host01` system. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- You have opened a terminal window.

Note

In this practice, remote connections and transfers of files will occur between two Oracle Linux systems (`host01 and host02`), where `host02` acts as the remote system. Take care to note on which system you are being asked to carry out a command as different tasks require specific commands to be run on a specific system. Both systems have the `ssh` service enabled to perform remote connections and secure file transfers.

Before You Begin

Ensure the following:

- Both the `host01` and `host02` systems are up and running and you have their access credentials. (See *Course Practice Environment: Security Credentials* for the `oracle` user password.)
- Determine the IP Address of the `host01` and `host02` systems for intercommunication using any one of the following options:
  - Use the `ifconfig -a` command.

  For example:

```
[oracle@host01]$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.237.16.181  netmask 255.255.254.0  broadcast
10.237.17.255
        inet6 fe80::a8bb:ff:fe01:8421  prefixlen 64  scopeid
0x20<link>
        ether aa:bb:00:01:84:21  txqueuelen 1000  (Ethernet)
        RX packets 443476  bytes 416838600 (397.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 146893  bytes 1065836109 (1016.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
0
…
```

- Refer to the entries made in the `/etc/hosts` file.

For example:

```
[oracle@host01]$ cat /etc/hosts
…
10.237.16.181 host01.us.oracle.com host01 eg18421
```

Tasks

1. Launch the gnome calculator on the remote Oracle Linux system (`host02`) using the `gnome-calculator` command. This is required for a later task in this practice where you will attempt to perform a command on a remote system to kill the process related to this calculator tool.

   a. In your remote Oracle Linux system (`host02`), open a terminal window and run the `gnome-calculator` command. Leave the calculator tool open.

      host02

      ```
      [oracle@host02]$ gnome-calculator
      ```


   Note: You may ignore the warning messages that show up in the terminal window.

2. Open a terminal on your local Oracle Linux system (`host01`) and use the `ssh` command to log on to the remote Oracle Linux system (`host02`) in your virtual network.

host01

```
[oracle@host01]$ ssh oracle@<IP address of host02>
...
ECDSA key fingerprint is
SHA256:K2wnJmhSnhEyxVYl4X676NkQWx6uMOvnbu/XP6Msppw.
Are you sure you want to continue connecting
(yes/no/[fingerprint])? yes
...
oracle@<IP address of host02>'s password: <Password>
...
[oracle@host02]$ pwd
/home/oracle
[oracle@host02]$
```

Or

host01

```
[oracle@host01]$ ssh -l oracle <IP address of host02>
...
ECDSA key fingerprint is
SHA256:K2wnJmhSnhEyxVYl4X676NkQWx6uMOvnbu/XP6Msppw.
Are you sure you want to continue connecting
(yes/no/[fingerprint])? yes
...
oracle@<IP address of host02>'s password: <Password>
...
[oracle@host02]$ pwd
/home/oracle
[oracle@host02]$
```

Note: The default directory on the remote machine will be the user's home directory `/home/oracle`.

3. While logged on to the remote Oracle Linux system (host02) from the local Oracle Linux system (host01), perform the following steps:

a. Use the uname command to display the host name of the current system.

host01

```
[oracle@host02]$ uname -n
host02
[oracle@host02]$
```

b. Use the ps command to identify the PID of the gnome-calculator command on the remote system.

host01

```
[oracle@host02]$ ps -ef | grep gnome-calculator
oracle    135190  135157  0 02:40 pts/0    00:00:00 gnome-
calculator
oracle    135525  135403  0 02:58 pts/2    00:00:00 grep --
color=auto gnome-calculator
[oracle@host01]$
```

Note: The first entry shown with a PID of 135190 is the process ID of the Calculator tool on the remote system. The second entry represents the ps command using grep to list the process with the name gnome-calculator.

c. Terminate the gnome-calculator process using the kill command and the process PID.

host01

```
[oracle@host01]$ kill 135190
```

Note:

- If you log on to the remote system as the root user, you can terminate the process.

- If you log on to the remote system as the same user (same UID) as the user who started the process on the remote system, then too you can terminate the process.

- However, if you log on to the remote system as some other user, you cannot terminate the process because you do not own the process and do not have the appropriate permission.

d. Log out of the remote system.

host01

```
[oracle@host02]$ exit
logout
...
[oracle@host01]$
```

4. Display the host name of your current system to determine whether you have returned to your local system.

host01

```
[oracle@host01]$ uname -n
host01
[oracle@host01]$
```

5. Next, use the scp command to copy a file from the remote Oracle Linux system (host02) to your local Oracle Linux system (host01).

a. Use the scp command to copy the dante file from the /home/oracle/lab directory on the remote Oracle Linux system (host02) to the /home/oracle/lab/dir1 directory on your local Oracle Linux system (host01).

host02

```
[oracle@host02]$ cd lab
[oracle@host02]$ ls -l dante*
-rwxrwxr-x. 1 oracle oracle 1319    Mar  5  2018 dante
-rwxrwxr-x. 1 oracle oracle  368    Mar  5  2018 dante_1
[oracle@host02]$


[oracle@host02]$ scp dante <IP address of
host01>:/home/oracle/lab/dir1
oracle@<IP address of host02>'s password: <Password>
dante                                100% 1319   116.5KB/s   00:00
[oracle@host01]$
```

b. Verify that the dante file has been copied to the /home/oracle/lab/dir1 directory on your local Oracle Linux system (host01).

host01

```
[oracle@host01]$ cd lab/dir1
[oracle@host01]$ ls -l dante*
-rwxrwxr-x. 1 oracle oracle 1319 Dec 14 03:27 dante
[oracle@host01]$
```

6. Now, copy the dante file from your local Oracle Linux system (host01) to the lab/dir2 directory on your remote Oracle Linux system (host02).

a. Check that the dante file is not already in the remote system directory.

host02

```
[oracle@host02]$ cd /home/oracle/lab/dir2
[oracle@host02]$ ls -l dante*
ls: cannot access 'dante*': No such file or directory
[oracle@host02]$
```

b. Copy the file using the `scp` command.

host01

```
[oracle@host01]$ scp dante <IP address of
host02>:/home/oracle/lab/dir2
oracle@<IP address of host02>'s password: <Password>
dante                         100% 1319      1.8MB/s   00:00
[oracle@host01]$
```

c. Verify that the `dante` file has been copied.

host02

```
[oracle@host02]$ ls -l dante*
-rwxrwxr-x. 1 oracle oracle 1319 Dec 14 08:18 dante
[oracle@host02]$
```

7. Copy the `lab/practice` directory from your remote Oracle Linux system (`host2`) to your home directory on your local Oracle Linux system (`host01`).

a. On your remote Oracle Linux system (`host02`), return to the `lab` directory, and use the `scp -r` command to copy the `practice` directory and its contents to your local Oracle Linux system (`host01`).

host02

```
[oracle@host02]$ cd ..
[oracle@host02]$ scp -r practice <IP address of
host01>:/home/oracle
oracle@<IP address of host01>'s password: <Password>
mailbox                                       100%    0
0.0KB/s   00:00
results                                       100%    0
0.0KB/s   00:00
project                                       100%    0
0.0KB/s   00:00
projection                                    100%    0
0.0KB/s   00:00
research                                      100%    0
0.0KB/s   00:00
[oracle@host02]$
```

b. Check the directory and its contents in your home directory on your remote system.

host01

```
[oracle@host01]$ cd
[oracle@host01]$ ls -l
total 176
-rw-rw-r--.  1 oracle oracle      0 Nov 26 11:41 celery
drwxr-xr-x.  2 oracle oracle   4096 Sep  3 14:09 Desktop
drwxr-xr-x.  2 oracle oracle   4096 Sep  1 20:37 Documents
drwxr-xr-x.  2 oracle oracle   4096 Nov 25 09:52 Downloads
drwxrwxrwx.  2 root   root     4096 Nov 26 06:24 eKit
-rw-rw-r--.  1 oracle oracle     44 Nov 27 06:44 example
drwxrwxr-x.  2 oracle oracle   4096 Nov 26 11:42 house
drwxrwxr-x. 14 oracle oracle   4096 Nov 27 10:37 lab
-rw-rw-r--.  1 oracle oracle 112640 Nov 27 10:23 lab.tar
drwxr-xr-x.  2 oracle oracle   4096 Sep  1 20:37 Music
drwxr-xr-x.  2 oracle oracle   4096 Sep  1 20:37 Pictures
drwxrwxr-x.  2 oracle oracle   4096 Dec 14 08:35 practice
drwxr-xr-x.  2 oracle oracle   4096 Sep  1 20:37 Public
drwxrwxr-x.  2 oracle oracle   4096 Nov 26 11:41 records
drwxrwxr-x.  2 oracle oracle   4096 Nov 27 05:54 sbin
drwxr-xr-x.  2 oracle oracle   4096 Sep  1 20:37 Templates
drwxrwxr-x.  2 oracle oracle   4096 Nov 26 11:43 veggies
drwxr-xr-x.  2 oracle oracle   4096 Sep  1 20:37 Videos

[oracle@host01]$ ls -l practice
total 0
-rwxrwxr-x. 1 oracle oracle 0 Dec 14 08:35 mailbox
-rwxrwxr-x. 1 oracle oracle 0 Dec 14 08:35 project
-rwxrwxr-x. 1 oracle oracle 0 Dec 14 08:35 projection
-rwxrwxr-x. 1 oracle oracle 0 Dec 14 08:35 research
-rwxrwxr-x. 1 oracle oracle 0 Dec 14 08:35 results
[oracle@host01]$
```

8. Use the `sftp` command to securely retrieve the file, `myvars`, from the `lab` directory on your local Oracle Linux system (`host01`) to the `/home/oracle` directory on your remote Oracle Linux system (`host02`).

    host02

    ```
    [oracle@host02]$ cd
    [oracle@host02]$ sftp <IP address of host01>
    oracle@<IP address of host01>'s password: <Password>
    Connected to <IP address of host01>.
    sftp> pwd
    Remote working directory: /home/oracle
    sftp> get lab/myvars
    Fetching /home/oracle/lab/myvars to myvars
    /home/oracle/lab/myvars              100%   67     28.3KB/s   00:00
    sftp> exit
    [oracle@host02]$ ls -l myvars
    -rwxrwxr-x. 1 oracle oracle 67 Dec 14 08:45 myvars
    [oracle@host02]$
    ```

    Note: You can also exit the sftp session with the `quit` or `bye` command at the `sftp>` prompt.

9. Using the string "file", transfer multiple files from the `lab` directory on your local Oracle Linux system (`host01`) to the `/home/oracle` directory on your remote Oracle Linux system (`host02`).

    a. Check that the files are not already there in the home directory of your remote Oracle Linux system (`host02`).

    host02

    ```
    [oracle@host02]$ ls -l
    total 128
    drwxr-xr-x. 2 oracle oracle  4096 Sep  3 14:09 Desktop
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Documents
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Downloads
    drwxrwxrwx. 2 root   root    4096 Dec  3 09:36 eKit
    drwxr-xr-x. 7 oracle oracle  4096 Dec 14 12:26 lab
    -rw-rw-r--. 1 oracle oracle 81920 Dec 14 12:33 lab.tar
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Music
    -rwxrwxr-x. 1 oracle oracle    67 Dec 14 08:45 myvars
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Pictures
    drwxrwxr-x. 2 oracle oracle  4096 Dec 14 08:34 practice
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Public
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Templates
    drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Videos
    [oracle@host02]$
    ```

b. Establish an sftp session to your local Oracle Linux system (`host01`) and retrieve all
    files with the string "file" in their name.

    host02

```
[oracle@host02]$ sftp <IP address of host01>
oracle@<IP address of host01>'s password: <Password>
Connected to <IP address of host01>.
sftp> pwd
Remote working directory: /home/oracle
sftp> cd lab
sftp> pwd
Remote working directory: /home/oracle/lab
sftp> mget file*
Fetching /home/oracle/lab/file.1 to file.1
Fetching /home/oracle/lab/file.2 to file.2
Fetching /home/oracle/lab/file.3 to file.3
Fetching /home/oracle/lab/file1 to file1
/home/oracle/lab/file1               100% 1610    94.7KB/s   00:00
Fetching /home/oracle/lab/file2 to file2
/home/oracle/lab/file2               100%  105    56.3KB/s   00:00
Fetching /home/oracle/lab/file3 to file3
/home/oracle/lab/file3               100%  218    21.9KB/s   00:00
Fetching /home/oracle/lab/file4 to file4
/home/oracle/lab/file4               100%  137    83.2KB/s   00:00
sftp> exit
[oracle@host02]$ ls -l
total 144
drwxr-xr-x. 2 oracle oracle  4096 Sep  3 14:09 Desktop
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Documents
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Downloads
drwxrwxrwx. 2 root   root    4096 Dec  3 09:36 eKit
-rwxrwxr-x. 1 oracle oracle     0 Dec 14 12:37 file.1
-rwxrwxr-x. 1 oracle oracle  1610 Dec 14 12:37 file1
-rwxrwxr-x. 1 oracle oracle     0 Dec 14 12:37 file.2
-rwxrwxr-x. 1 oracle oracle   105 Dec 14 12:37 file2
-rwxrwxr-x. 1 oracle oracle     0 Dec 14 12:37 file.3
-rwxrwxr-x. 1 oracle oracle   218 Dec 14 12:37 file3
-rwxrwxr-x. 1 oracle oracle   137 Dec 14 12:37 file4
drwxr-xr-x. 7 oracle oracle  4096 Dec 14 12:26 lab
-rw-rw-r--. 1 oracle oracle 81920 Dec 14 12:33 lab.tar
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Music
-rwxrwxr-x. 1 oracle oracle    67 Dec 14 08:45 myvars
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Pictures
```

```
drwxrwxr-x. 2 oracle oracle  4096 Dec 14 08:34 practice
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Public
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Templates
drwxr-xr-x. 2 oracle oracle  4096 Sep  1 20:37 Videos
[oracle@host02]$
```

10. Transfer the myvars file from the /home/oracle directory on your remote Oracle Linux system (host02) to the /home/oracle directory on your local Oracle Linux system (host01).

    a. Establish the session with sftp and check that the myvars file is not already there in the remote system directory.

host02

```
[oracle@host02]$ sftp <IP address of host01>
oracle@<IP address of host01>'s password: <Password>
Connected to <IP address of host01>.
sftp> ls -l
drwxr-xr-x    2 oracle   oracle        4096 Sep  3 14:09 Desktop
drwxr-xr-x    2 oracle   oracle        4096 Sep  1 20:37
Documents
drwxr-xr-x    2 oracle   oracle        4096 Nov 25 09:52
Downloads
drwxr-xr-x    2 oracle   oracle        4096 Sep  1 20:37 Music
drwxr-xr-x    2 oracle   oracle        4096 Sep  1 20:37 Pictures
drwxr-xr-x    2 oracle   oracle        4096 Sep  1 20:37 Public
drwxr-xr-x    2 oracle   oracle        4096 Sep  1 20:37
Templates
drwxr-xr-x    2 oracle   oracle        4096 Sep  1 20:37 Videos
-rw-rw-r--    1 oracle   oracle           0 Nov 26 11:41 celery
drwxrwxrwx    2 root     root          4096 Nov 26 06:24 eKit
-rw-rw-r--    1 oracle   oracle          44 Nov 27 06:44 example
drwxrwxr-x    2 oracle   oracle        4096 Nov 26 11:42 house
drwxrwxr-x   14 oracle   oracle        4096 Nov 27 10:37 lab
drwxrwxr-x    2 oracle   oracle        4096 Dec 14 08:35 practice
drwxrwxr-x    2 oracle   oracle        4096 Nov 26 11:41 records
drwxrwxr-x    2 oracle   oracle        4096 Nov 27 05:54 sbin
drwxrwxr-x    2 oracle   oracle        4096 Nov 26 11:43 veggies
sftp>
```

b. Use the `put` command to transfer the `myvars` file to your local Oracle Linux system (`host01`).

host02

```
sftp> put myvars
Uploading myvars to /home/oracle/myvars
myvars                              100%   67      79.3KB/s    00:00
sftp> ls -l
drwxr-xr-x    2 oracle    oracle       4096 Sep  3 14:09 Desktop
drwxr-xr-x    2 oracle    oracle       4096 Sep  1 20:37
Documents
drwxr-xr-x    2 oracle    oracle       4096 Nov 25 09:52
Downloads
drwxr-xr-x    2 oracle    oracle       4096 Sep  1 20:37 Music
drwxr-xr-x    2 oracle    oracle       4096 Sep  1 20:37 Pictures
drwxr-xr-x    2 oracle    oracle       4096 Sep  1 20:37 Public
drwxr-xr-x    2 oracle    oracle       4096 Sep  1 20:37
Templates
drwxr-xr-x    2 oracle    oracle       4096 Sep  1 20:37 Videos
-rw-rw-r--    1 oracle    oracle          0 Nov 26 11:41 celery
drwxrwxrwx    2 root      root         4096 Nov 26 06:24 eKit
-rw-rw-r--    1 oracle    oracle         44 Nov 27 06:44 example
drwxrwxr-x    2 oracle    oracle       4096 Nov 26 11:42 house
drwxrwxr-x   14 oracle    oracle       4096 Nov 27 10:37 lab
-rwxrwxr-x    1 oracle    oracle         67 Dec 14 12:43 myvars
drwxrwxr-x    2 oracle    oracle       4096 Dec 14 08:35 practice
drwxrwxr-x    2 oracle    oracle       4096 Nov 26 11:41 records
drwxrwxr-x    2 oracle    oracle       4096 Nov 27 05:54 sbin
drwxrwxr-x    2 oracle    oracle       4096 Nov 26 11:43 veggies
sftp> exit
[oracle@host02]$
```

Practices for Lesson 9: Archiving Files and Remote File Transfers

11. Using the `mput` command, transfer multiple files starting with the string "file" from the /home/oracle directory on your remote Oracle Linux system (host02) to the /home/oracle directory on your local Oracle Linux system (host01).

host02

```
[oracle@host02]$ sftp <IP address of host01>
oracle@<IP address of host01>'s password: <Password>
Connected to <IP address of host01>.
sftp> pwd
Remote working directory: /home/oracle
sftp> mput file*
Uploading file.1 to /home/oracle/file.1
file.1                          100%    0    0.0KB/s   00:00
Uploading file.2 to /home/oracle/file.2
file.2                          100%    0    0.0KB/s   00:00
Uploading file.3 to /home/oracle/file.3
file.3                          100%    0    0.0KB/s   00:00
Uploading file1 to /home/oracle/file1
file1                           100% 1610    2.0MB/s   00:00
Uploading file2 to /home/oracle/file2
file2                           100%  105  186.0KB/s   00:00
Uploading file3 to /home/oracle/file3
file3                           100%  218  229.3KB/s   00:00
Uploading file4 to /home/oracle/file4
file4                                        100%   137
32.2KB/s   00:00
sftp> ls -l
drwxr-xr-x    2 oracle    oracle        4096 Sep  3 14:09 Desktop
drwxr-xr-x    2 oracle    oracle        4096 Sep  1 20:37
Documents
drwxr-xr-x    2 oracle    oracle        4096 Nov 25 09:52
Downloads
drwxr-xr-x    2 oracle    oracle        4096 Sep  1 20:37 Music
drwxr-xr-x    2 oracle    oracle        4096 Sep  1 20:37 Pictures
drwxr-xr-x    2 oracle    oracle        4096 Sep  1 20:37 Public
drwxr-xr-x    2 oracle    oracle        4096 Sep  1 20:37
Templates
drwxr-xr-x    2 oracle    oracle        4096 Sep  1 20:37 Videos
-rw-rw-r--    1 oracle    oracle           0 Nov 26 11:41 celery
drwxrwxrwx    2 root      root          4096 Nov 26 06:24 eKit
-rw-rw-r--    1 oracle    oracle          44 Nov 27 06:44 example
-rwxrwxr-x    1 oracle    oracle           0 Dec 14 12:46 file.1
-rwxrwxr-x    1 oracle    oracle           0 Dec 14 12:46 file.2
-rwxrwxr-x    1 oracle    oracle           0 Dec 14 12:46 file.3
```

```
-rwxrwxr-x     1 oracle    oracle        1610 Dec 14 12:46 file1
-rwxrwxr-x     1 oracle    oracle         105 Dec 14 12:46 file2
-rwxrwxr-x     1 oracle    oracle         218 Dec 14 12:46 file3
-rwxrwxr-x     1 oracle    oracle         137 Dec 14 12:46 file4
drwxrwxr-x     2 oracle    oracle        4096 Nov 26 11:42 house
drwxrwxr-x    14 oracle    oracle        4096 Nov 27 10:37 lab
-rwxrwxr-x     1 oracle    oracle          67 Dec 14 12:43 myvars
drwxrwxr-x     2 oracle    oracle        4096 Dec 14 08:35 practice
drwxrwxr-x     2 oracle    oracle        4096 Nov 26 11:41 records
drwxrwxr-x     2 oracle    oracle        4096 Nov 27 05:54 sbin
drwxrwxr-x     2 oracle    oracle        4096 Nov 26 11:43 veggies
sftp> exit
[oracle@host02]$
```

12. Close all the open terminal windows.